VISUALIZING EARLY-STAGE BREAST CANCER TUMORS IN A MAMMOGRAPHIC ENVIRONMENT THROUGH A 3-DIMENSIONAL MATHEMATICAL MODEL

C. Brian Bassham, First Lieutenant, USAF

AFIT/GOA/ENS/99M-01

AFIT/GOA/ENS/99M-01

VISUALIZING EARLY-STAGE BREAST CANCER TUMORS IN A MAMMOGRAPHIC ENVIRONMENT THROUGH A 3-DIMENSIONAL MATHEMATICAL MODEL

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

C. Brian Bassham, B.S.

First Lieutenant, USAF

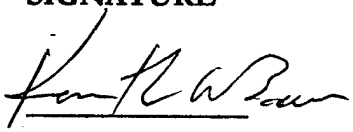
MARCH, 1999

THESIS APPROVAL

NAME: C. Brian Bassham, First Lieutenant, USAF **CLASS:** GOA-99M

THESIS TITLE: Visualizing Early-stage Breast Cancer Tumors in a Mammographic Environment Through a 3-Dimensional Mathematical Model

DEFENSE DATE: 05 MARCH, 1999

COMMITTEE:	**NAME/TITLE/DEPARTMENT**	**SIGNATURE**
Advisor	Kenneth Bauer, Dr. Professor Department of Operational Sciences	
Reader	John O. Miller, Lieutenant Colonel, USAF Assistant Professor Department of Operational Sciences	
Additional Readers	Delano D. Wilson, Major, USAF, MC, SFS Lead Aeromedical Research Physician Air Force Research Laboratory	

Acknowledgments

I would like to first thank my God for allowing me to achieve the goals that I have set for myself, to see the beauty in life, and to merely exist. Without His help, I would be nothing. Not my will, but Thine.

I would like to express my love and devotion to my lovely wife, Bonnie, and my wonderful son, Colby. They have been patient and overwhelmingly loving through this adventure. They are my favorite distractions.

I must thank my advisor and primary reader, Dr. Ken Bauer and Lt. Col. J. O. Miller. Their remarkable insight into problems and friendliness have made this thesis effort an absolute joy. I would also like to show my appreciation for the help offered by the rest of the thesis committee: Maj. Del Wilson, Dr. Jeff Hoffmeister, Dr. Steve Rogers, and Dr. Matthew Kabrisky. They bolstered this work by bringing many exciting ideas to the table and adding their invaluable and overwhelming knowledge of breast cancer! Finally, I wish to thank my friend, Dr. Andy Chunn, from whom I continuously sought medical explanation. Maybe one of these days I will know as much as he does!

Brian Bassham

Table of Contents

List of Figures

List of Figures (cont.)

Figure **Page**

List of Tables

Abstract

In response to the insidious and deadly nature of breast cancer and the less-than-perfect detection ability of mammography, we develop a mathematical model as a foundation to the long-term goal of improving early breast cancer detection. By using modeling and simulation to construct an accurate breast cancer tumor model, we hope to solve the problems associated with mammogram misdiagnosis and, perhaps as a by-product, lend insight to tumor development dynamics. The final tumor model, written in MATLAB, provides realistic tumor growth and 2-dimensional visualization of 3-dimensional structures. Earlier modeling attempts capture *slices* of the tumor in the 2-dimensional growth spaces. The final 3-dimensional model closely mimics the characteristics of theoretical breast cancer development within the female breast by establishing an algorithm that reliably represents the *ideal tumor model.* The possible impact of this model and its progeny is earlier detection of breast cancer which leads to an increased chance of survival for those afflicted with the disease.

VISUALIZING EARLY-STAGE BREAST CANCER TUMORS IN A MAMMOGRAPHIC ENVIRONMENT THROUGH A 3-DIMENSIONAL MATHEMATICAL MODEL

I. Introduction.

1.1 Background.

Breast cancer is the "single most common cancer diagnosed in women in the United States" and accounts for about 20% of cancer deaths among females [AMWA, 1996:3]. Despite the presence of this ominous killer, little ground has been gained in the early treatment of this disease [Cotran et al, 1989:1192]. The only "cures" available are massive tissue removal through mastectomies or radiotherapy, which can be dangerous and aesthetically unappealing [Harris et al, 1991:347-370].

Researchers and other medical professionals have noted a deficiency in current cancer detection techniques, specifically those used to detect the tumors that define breast cancer [87]. Theoretically, by the time a tumor is able to be detected by touch, or palpable, it has already spent three-quarters of its life-span living in its host [Zander and Baltzer, 1985:84]. Therefore, a tumor must be removed before it has a chance to spread to other parts of the body, or metastasize; a capability that the tumor may have already accomplished by the time that it is palpable [84]. In fact, if a woman's cancer is found "when it is smaller than 1.0 cm in diameter and has not spread to the lymph nodes, it can be treated with a simple operation that leaves the breast intact and she has a 95% chance of being cured" [Olivotto et al, 1995:33]. In another light, distant metastases of breast cancer to other areas

of the body, predominately the bone, liver, and lungs, is the primary means by which breast cancer kills its host [Harris et al, 1991:178][Kopans, 1998:36]. Therefore, time is a major factor in the removal of malignant tumors from the body, especially before they have a chance to spread. Due to the absence of a true cure and breast cancer's highly sporadic and oftentimes rapid growth patterns, "the diagnosis of breast cancer at an early stage in its development must remain the measure most likely to reduce the mortality of the disease." [Brunner et al, 1984:173].

1.2 Problem Statement.

Several problems may arise when medical professionals screen for breast cancer. The first steps in the breast cancer screening process, a self-examination and a physical examination by a doctor, are not guaranteed to lead to a positive detection of an existing tumor [Harris et al, 1991:87]. Tumors may not be palpable until long after their existence in the body. Therefore, a breast self-examination (BSE) or a physical examination by a trained physician offers little help in the earliest stages of breast cancer. Thus, a better detection technique, like mammography, is required for finding evidence of breast cancer.

Despite their advantage of detection sensitivity over physical breast examinations, mammograms have weaknesses also. Medical personnel operating mammographic machinery may not properly situate the breast when taking x-ray images, while doctors examining mammograms may easily miss the evidence of breast cancer by not knowing exactly what to detect. Since mammograms cannot distinguish malignant tumors from other benign lesions, medical professionals must be well-trained on identifying characteristics typical of breast cancer in order to know if, when, and how to combat the suspected lesion

[88]. Mammographic tumor shapes, shadows, and other small evidences of breast cancer, such as microcalcifications or areas of variable tissue density, must be readily recognizable. Highly random tumor growth rates can cause malignant tumors to remain undetected, incur widely variable tumor growth patterns, and allow the tumors to spread to various parts of the body very quickly. A final problem in the search for breast cancer is the low percentage of palpable breast "lumps" that are indeed malignant tumors. In a study of 1,000 patients at a surgical outpatient department, only 10% of the women who had detected palpable lumps actually had "biopsy-proven cancer" [Cotran et al, 1989:1183]. This low percentage of occurrence indicates that cancer can be a "needle in a haystack" from the very beginning. Figure 1-1 depicts the overall breakdown of the breast "lumps" observed in the study.

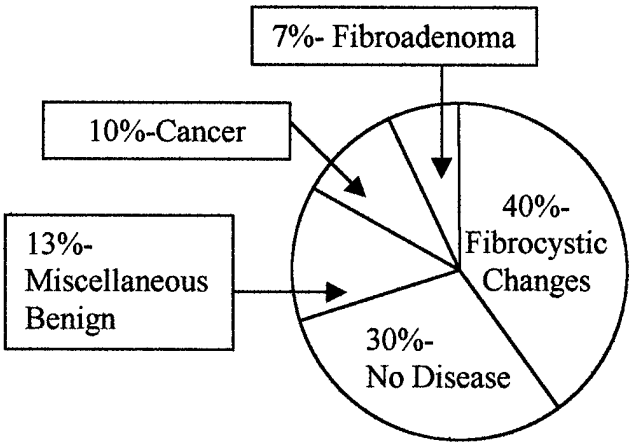


Figure 1-1. Classification of Breast "Lumps" in a Surgical Outpatient Department Study of 1000 women. [Cotran, 1989]

1.3 Research Objectives and Questions.

In the preface to their collection of essays concerning mathematical tumor behavior models, John Adam and Nicola Bellomo acknowledge that mathematics and computer science may be able to make major contributions to problems encountered by biologists, immunologists, and all attempting to solve the problems associated with cancer, but that this realization is distant [Adam and Bellomo, 1997:xi]. They contend that mathematical models and computer simulations may decompose the complex cancer system into relatively simple elements that may emulate the natural behavior of the cancer [xi]. Current cancer research and tumor modeling has been predominately devoted to observing how cancer cells interact with various chemicals, medicines, hormones, and steroids. In other words, research has primarily been devoted to hunting for a cure. Many mathematical models used today, though useful, offer little insight into the visual aspects of the tumors because they assume an ideally spherical shape and usually focus on the growth rate aspects and cellular-hormonal interactions of malignant tumors. However, a mathematical model that accurately simulates tumor growth and has the ability to visualize the tumor development could conceivably offer insight to the problems associated with finding breast cancer.

Thus, the objective is to attempt to model the growth of cancer from its very beginning -- as a single cell. Beginning the modeling process when a tumor becomes palpable is far too late to be helpful. A mathematical model that can reliably grow, form, and metastasize like a real-world breast cancer tumor would be invaluable in understanding where, when, and how to search for these killer cells. Many questions could be explored through the mathematical modeling of the breast cancer tumors, including:

1.) What makes a cancer tumor look distinctive at the earliest stages?

2.) What does a tumor look like from several different angles?

3.) Can the randomness of cancer be statistically quantified?

4.) What influence do other breast structures have upon the physical shaping of breast cancer?

1.4 Proposed Solution.

By producing a 3-dimensional representation of a given tumor through mathematical modeling and subjecting it to physical forces, such as pressure from surrounding breast structures, and x-ray radiation from mammographic detection machinery, a better glimpse of how these tumors appear during the early stages of their life cycle may be attained. Armed with a reliable concept of cancer's visual characteristics provided by a mathematical model, doctors may be able to improve their success rate in the early detection of malignant tumors. Figure 1-2 presents a graphical stepwise process of how a successful modeling plan may be accomplished.

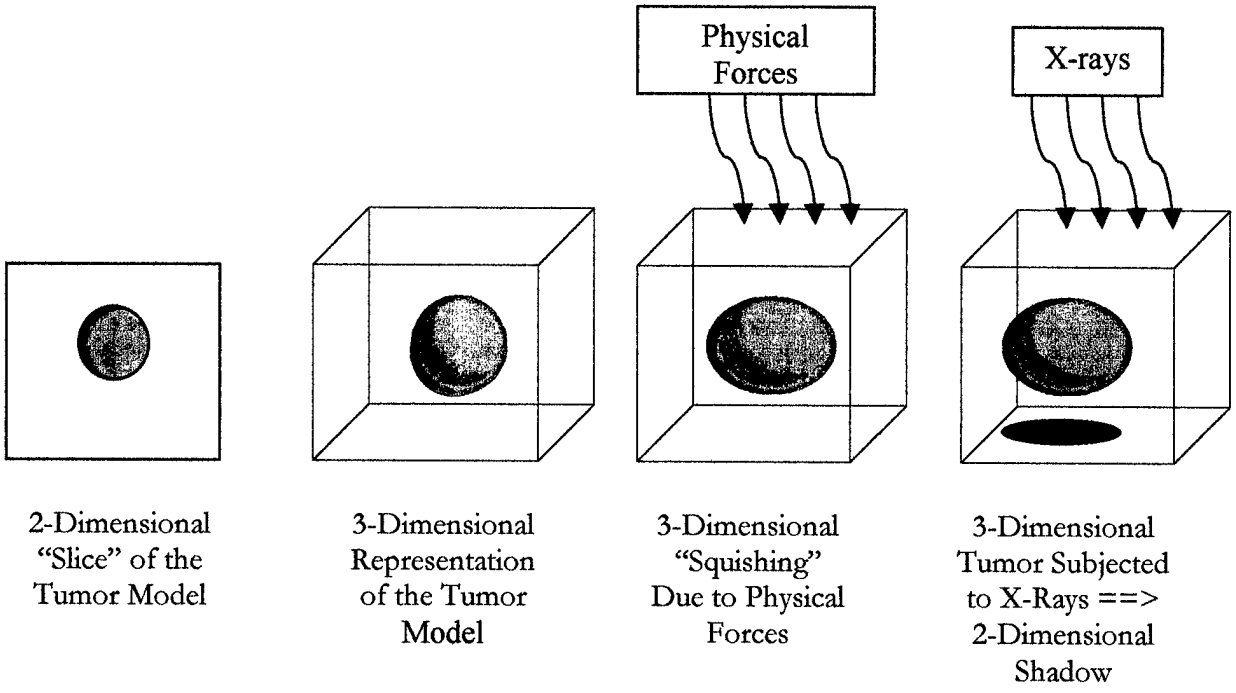


Figure 1-2. Overall Modeling Methodology.

1.5 Research Scope.

The scope of this thesis research deals primarily with breast cancer, but may be applied to other cancer when appropriate. The research does not focus on the intricacies of breast cancer, such as gene inheritance, risk factors, and reasons for occurrence, but rather with the information surrounding the theoretical growth sequences of the tumor on the cellular level and its visual representation. However, more in-depth aspects of breast cancer may be introduced through subsequent thesis work.

II. Literature Review

2.1 Background.

While the solution to the problem of modeling breast cancer tumors may eventually require total knowledge of the intricacies of breast cancer and its effects, the beginning steps of the modeling process requires only a firm understanding of the basic structures, locations, and dynamics of breast cancer, as well as the limitations of current screening and detection techniques. By no means exhaustive, this literature review merely gives a glimpse into the complex nature of malignant tumor research and general topics that are encountered. The bulk of the literary search was conducted at the Fordham Medical Library on the Wright State University campus. Other valuable material originated from subject searches on the Internet and from meetings with knowledgeable personnel from the Air Force Institute of Technology, the Air Force Research Laboratory, Qualia Computing, Inc., and surrounding area hospitals.

An overview of the information contained within the literature review will provide a focus for the understanding of how breast cancer works and where the process of modeling the tumors should begin. First, knowledge of the structures of the breast is key to understanding where breast cancer begins, how it is initially contained and reacts with surrounding tissue, and how the tumors transform in shape as they grow. Next, an idea of breast cancer characteristics, behavior, and randomness must be grasped. Then, information on how mammography works and what criteria medical personnel use to detect breast cancer is key to the visualization aspects of modeling breast cancer. Finally, a short study of

current mathematical tumor modeling techniques will be beneficial to the modeling methodology and will ensure that work is not duplicated.

2.2 The Breast Structure.

The inner female breast is composed of four primary structures: the lobes, the ducts, fat, and connective tissue [BICHS, 1997]. The overall distribution of these breast structures can be seen in Figure 2-1. The inner structure of the female breast can best be visualized by comparing it to a cluster of grapes [BIHCS, 1997]. The grapes themselves represent the lobes, which are distributed in a wheel spoke pattern emanating from the nipple-areolar complex within the breast. On average, there are 15 to 20 lobes in each female breast [BIHCS, 1997]. The lobes are actually collections of smaller structures called the lobules. Hundreds of thousands of these lobules exist in the average female breast [Olivotto et al, 1995:5]. These lobules contain the parenchyma, where milk is formed and secreted [AMWA, 1996:8]. Connected to the lobes are ducts, represented by the branch-like stems of the grape cluster. This structure is of particular interest because over 90% of breast carcinomas begin in the ducts of the female breast [Cotran et al, 1989:1194]. The primary purpose of the ducts is to transport milk out of the breast. The ducts extend from the lobes and grow in size as they connect to each other. Figure 2-2 depicts a single duct and how its many branches connect to the lobules. The large ducts empty into the lactiferous sinuses, which collect the milk near the nipple-areolar complex. Finally, milk is expressed through the nipple, which may have as many as 20 small openings [AMWA, 1996:8-9]. Surrounding the defined lobe and duct structures are the fat and connective tissue. The fat content of the breasts increases as the female body ages, especially after menopause [Olivotto et al, 1995:6].

This phenomenon becomes increasingly important when breast cancer detection techniques, such as mammography, are employed.

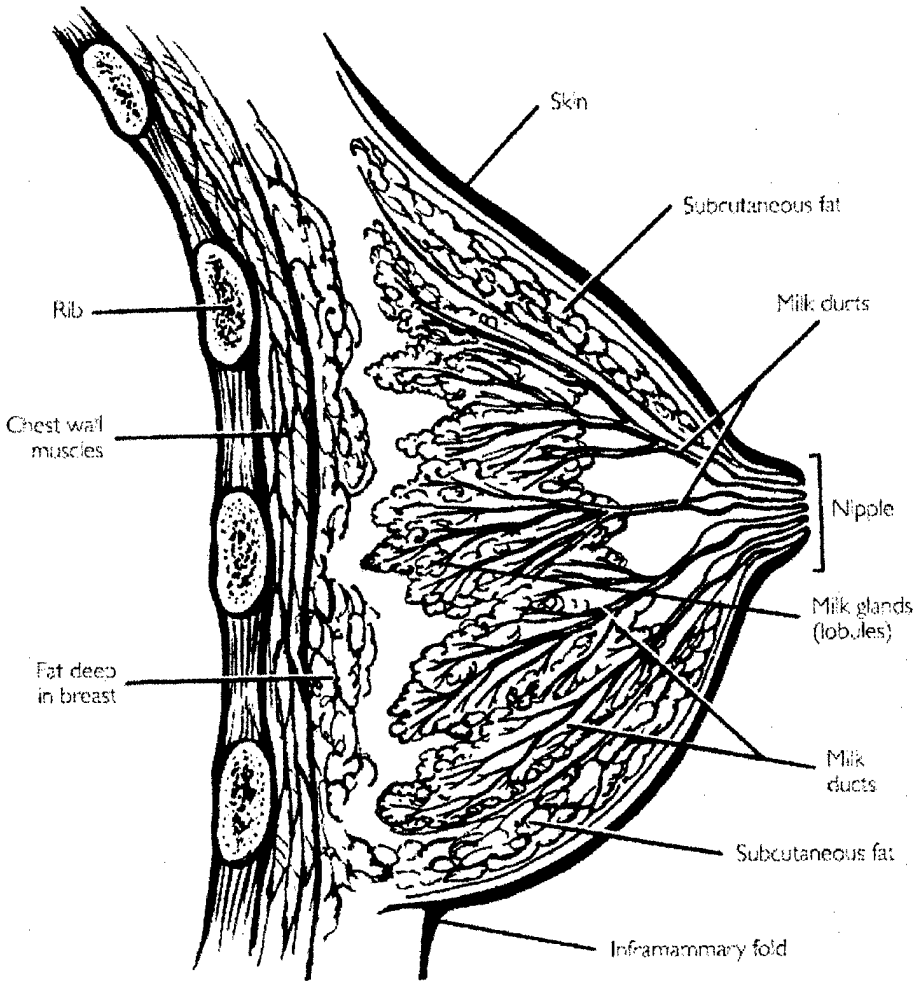


Figure 2-1. Female Breast Anatomy (Cross-section). [Olivotto, 1995]

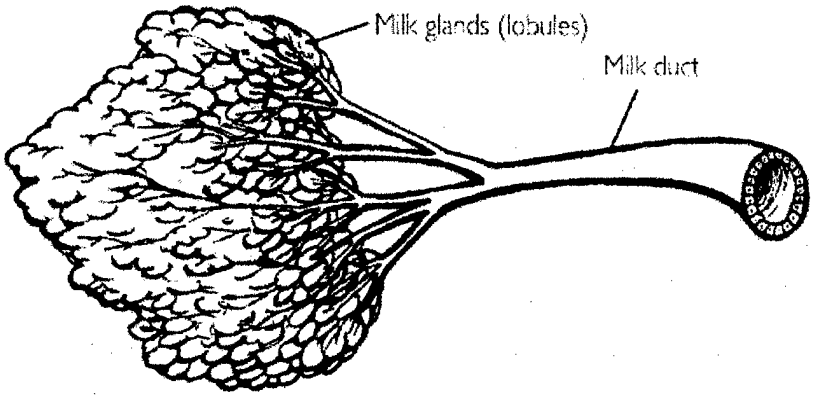


Figure 2-2. Female Breast Lobular and Ductal Layout. [Olivotto, p. 1995]

2.3 Breast Cancer.

The British oncologist Sir Rupert Willis defined neoplasia, cells of "new growth" that are typically called cancer when malignant, as "an abnormal mass of tissue, the growth of which exceeds and is uncoordinated with that of the normal tissues and persists in the same excessive manner after cessation of the stimuli which invoked the change" [Cotran et al, 1989:239]. To better understand how cancer cells behave, it is imperative to know how a normal breast cell behaves. Starting with a single, normal breast cell, the process begins when the cell begins to absorb nutrients provided by nearby blood vessels. After time passes, the cell divides, or mitoses, into two *daughter* cells. These daughter cells go on to divide in a like manner [Olivotto et al, 1995:13]. This growth cycle is very orderly and chemicals released by the body control the growth of the cells. Figure 2-3 illustrates this process.

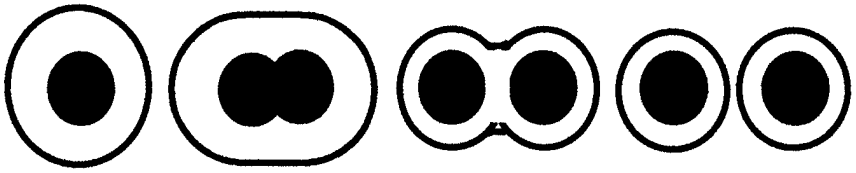


Figure 2-3. Normal Cell Division. [Olivotto, 1995]

Cancer cells, on the other hand, do not follow the signals released by the body nor do they behave in an organized manner like normal breast cells. There are three possible ways in which a cancer cell may develop. First, a normal breast cell may become malignant when it ceases to respond to growth-inhibiting signals and gains the ability to multiply uncontrollably [BIHCS, 1999]. Also, cancer cells may develop with the activation of growth-promoting oncogenes [Cotran et al, 1989:291]. Finally, normal cells may become cancerous when both incidences occur [291]. In other words, cancer may begin when cells no longer respond to the signals given by the body, when an outside agent transforms the cell, or both.

2.3.1 Cancer Growth Sequence.

The theoretical sequence of events for breast cancer growth and development is based upon the well-established sequence of cervical cancer. The assumption that this carries over to the breast is "more intuitive than real" [Zander and Baltzer, 1985:3]. For the sake of understanding the possible series of events relevant to this research, this assumption will be made. The theoretical growth and development sequence for breast cancer is shown in Figure 2-4:

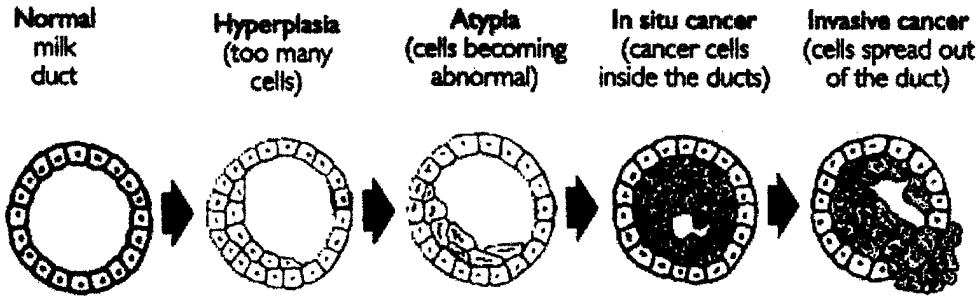


Figure 2-4. Breast Cancer Growth and Development Sequence. [Olivotto, 1995]

While this sequence presents the various stages that breast cancer can go through, it does not imply that breast cancer will always reach each stage. For example, an existing cancer may stay in situ for its entire existence or an invasive cancer may not be able to survive in its new environment [Kopans, 1998:36]. *Hyperplasia* refers to the process of growth, division, and rapid accumulation of cells. This occurs when the normal breast cells fail to respond to growth inhibitors and collect in the local area. *Atypia*, or atypical hyperplasia, refers to the growth, division, and accumulation of abnormal cells. In other words, the new cells no longer look like normal breast cells [Olivotto et al, 1995:16]. Cells that are of different sizes and shapes, or pleomorphic, are highly suggestive of carcinoma [AMWA, 1996:66]. *In situ* refers to the cancer in the breast that is still within the ducts or lobules; therefore, the cancer cells are in the same place, or *in* the *situ*ation, where they were first formed [Olivotto et al, 1995:76]. In situ cancer is not lethal and is generally debated whether it meets the definition of cancer; it is, however, a signaling precursor to cancer [Kopans, 1998:36]. Cancer that begins in the ducts, or ductal carcinoma in situ (DCIS), accounts for approximately 75% of all invasive breast cancers [Olivotto et al, 1995:81]. Also, as cancer cells collect within the duct and begin to block the passage of fluids, the chances for calcium deposits to occur

increase [81]. These small calcium deposits, or *microcalcifications*, are a possible sign of breast cancer [81]. Cancers that begin in the lobules, or lobular carcinoma in situ (LCIS), account for approximately 15% of all invasive breast cancers [81]. Finally, *invasive*, or infiltrating, cancer implies that the malignant cells have escaped from the confines of their original location and are now free to spread within the local area. Cancer is considered to be very dangerous when it reaches an invasive state. Breast cancer may then develop the properties that allow it to metastasize to other organs [Kopans, 1995:36]. When examining the effects of invasive carcinoma, the single most important factor in determining the future behavior of a tumor is the existence of breast cancer cells within the lymph nodes [Olivotto et al, 1995:8]. This evidence tells of the cancer's plan to strike the immune system and move to various organs of the body.

Another way to examine the breast cancer growth sequence is by determining the primary source of the tumor's nutrients. The two phases used in this method are the *prevascular*, or avascular, phase and the *vascular* phase. The prevascular phase begins with the creation of the first cancer cells. The tumor receives its nutrients and oxygen in the same way that the surrounding normal cells do, by diffusion [Adam et al, 1997:191]. Nutrients pass from distant blood vessels and through surrounding cells until they reach a cell that utilizes them. During the prevascular phase, a tumor grows very rapidly and soon begins to require more nutrients [188]. In fact, when the tumor does not receive enough nutrients, its innermost cells tend to die out [188]. This phenomenon, known as necrosis, is a typical characteristic of a rapidly growing tumor [Olivotto et al, 1995:14,75]. Nearing the end of this phase, the growth of the tumor slows as cells begin to conserve energy for survival. Unfortunately for the host, breast cancer tumors have the insidious ability to stimulate the

development of blood vessels around itself in order to feed its ravenous appetite for nutrients [Adam et al, 1997:188]. When the endothelial cells migrate towards and form capillary sprouts around the tumor, the vascular phase has begun. The tumor, now strengthened with its own personal nutrient supply, begins to grow rapidly once more. The prevascular phase is more representative of carcinoma in situ, while the vascular growth phase is generally typified by invasive cancer [188].

2.3.2 Cancer Distribution Within the Female Breast.

The location of breast cancer, when classified by breast quadrants, is typically distributed in the following sites: 50% in the upper outer quadrant, 10% in the upper inner quadrant, 10% in the lower outer quadrant, 10% in the lower inner quadrant, and 20% in the central, or subareolar, quadrant [Cotran et al, 1989:1193] [Harris et al, 1991:168]. This distribution is important when mammography is applied as a cancer detection technique. The spread of primary cancer through the breast occurs by infiltrating the parenchyma in the lobules, along the mammary ducts, and through the breast lymphatics [168]. This knowledge leads to the conclusion that with some likelihood the carcinoma will exist in places other than the palpable primary mass [168].

2.3.3 Breast Cancer Staging.

Due to its nearly unpredictable nature, breast cancer is difficult to categorize when *in vivo*, or inside the body. Once detected, there is really no way to know when the development of the cancer began or how fast the tumor is currently growing or spreading. Therefore, breast cancer is typically defined by the staging information given in Table 2-1. The staging technique relies on the fact that breast cancer tumors grow in size, spread to the lymph nodes, and metastasize to organs as they mature. There are additional ways to classify

breast cancer, such as the TNM (T, tumor; N, nodes; M, metastases) method, but are more complex than the Stage I, II, III, IV system [Harris et al, 1991:328].

Table 2-1. Breast Cancer Staging Criteria. [Olivotto, 1995] [AMWA, 1996]

Stage	Staging Criteria	Average 5-Year Survival
I	Tumor Size: ≤ 2 cm Lymph Nodes: Negative Metastases: None	80% to 95%
II	Tumor Size: > 2cm Lymph Nodes: Positive, but movable Metastases: None	50% to 70%
III	Tumor Size: Advanced Local (~5 cm) Lymph Nodes: Positive and fixed Metastases: None	30% to 60%
IV	Any Metastases Regardless of Tumor Size	5% to 20%

2.4 Mammography.

Of the various breast anomaly detection techniques, mammography is perhaps the most widely used. X-ray mammography is "a two-dimensional projection of three-dimensional structures and produces an image caused by the absorption and scattering of structures, based on their water content or, in calcifications, the absorption characteristics of the metals involved" [Harris et al, 1991:89]. Mammography is a detection tool that attempts to locate breast cancer tumors, along with other breast anomalies, by visualizing differences in the densities of tissue masses. The more dense tissue appears lighter on the x-ray film

than does the less dense tissue. The lighter image occurs when the x-rays attempt to pass through the patient's body, but are slowed or captured by the more dense tissue. The developed x-ray film is referred to as the mammogram. The dose of radiation from a mammogram is approximately 0.15 centigray (rads) [Olivotto et al, 1995:38].

It is important to note that mammography is not a diagnosis tool since its use cannot automatically differentiate between benign and malignant tumors [Harris et al, 1991:87]. In other words, questionable regions of concern on the mammogram may be misdiagnosed by the trained observer. Mammography can detect but cannot distinguish. Mammography's sensitivity, however, has been well documented [93]. Mammography has a false negative rate of about 15% [AMWA, 1996:48]. In other words, among women diagnosed for breast cancer, about 15 out of 100 women will be told they have a negative mammogram while having palpable breast cancer [48].

2.4.1 Visual Characteristics of Breast Cancer.

When reading a mammogram, a radiologist will examine the film for six common mammographic abnormalities: intramammary lymph nodes, smooth-walled masses, scattered microcalcifications, masses with irregular borders, clustered microcalcifications, and a *spiculated*,or spiky shape [AMWA, 1996:85]. The last three mammographic abnormalities typically distinguish breast cancer from other breast lesions. In fact, the primary sign that a mammogram lesion is indeed malignant is when the image depicts "ill-defined margins" or a spiculated shape [Harris et al, 1991:94]. With these qualities, the lesion is considered cancerous with 99% certainty [94]. The secondary signs of a malignant tumor are the clustered microcalcifications. These groupings appear as tiny white dots on the mammography film and may be the earliest indication of malignancy within the ducts [95].

By rule of thumb, five or more microcalcifications within a 1 cm area indicates possible malignancy and calls for further investigation [AMWA, 1996:87]. Figure 2-5a presents an actual mammogram where an atypical mass, not necessarily cancerous, has been detected. Another mammogram featured in Figure 2-5b depicts microcalcifications in a breast that contains ductal carcinoma in situ. Figure 2-5c shows a mammogram where microlobulated ductal carcinoma in situ has been detected along with microcalcifications. A malignant tumor can be seen in the mammogram featured in Figure 2-6. Note that the density on the left side in Figure 2-6 is not seen on the right side mammogram. This phenomenon indicates to a radiologist the presence of a breast anomaly, but does not automatically indicate malignancy [49].

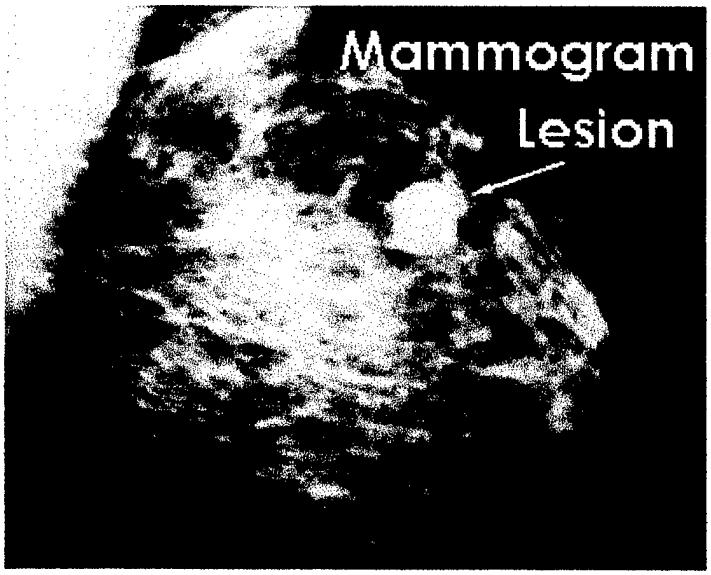


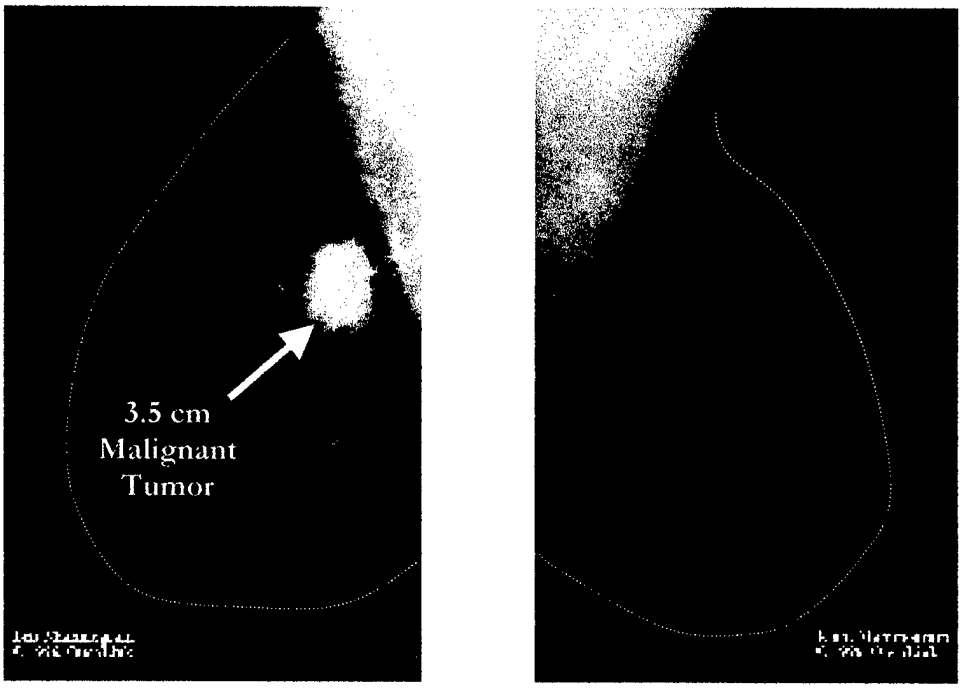
Figure 2-5a. Sample Mammogram. [Connelly, 1999]

Figure 2-5b. Sample Mammogram [University Hospitals of Cleveland, 1999].

Figure 2-5c. Sample Mammogram [University Hospitals of Cleveland, 1999]

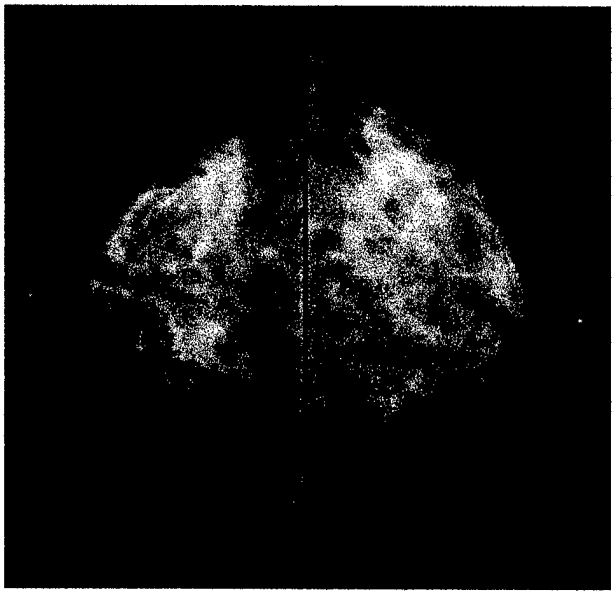
2.4.2 Mammography Viewpoints.

Since a single view of each breast reduces the probability of detecting an existing mass, two standard viewpoints of each breast are taken for each mammography examination. The cranio-caudal (C-C) perspective takes an overhead view as the X-ray beam enters the top of the breast and exits at a slight angle through the bottom of the breast [Harris et al, 1991:92]. The mediolateral oblique (MLO) view is basically a side view of the patient's breast in which the X-ray beam enters from the medial side and exposes film located next to the lateral surface of the breast [Harris et al, 1991:92] [AMWA, 1996:62]. Figures 2-6 and 2-7 illustrate the two standard views used. Additional perspectives can be taken in order to increase the chance of detecting a breast abnormality. The two standard views were established in order to maximize the exposure of breast quadrants where tumors are expected to occur and offer a standard set of viewpoints with which medical personnel are familiar.



Left Side Right Side

Figure 2-6. Mammograms (Mediolateral Oblique View). [University of Pennsylvania, 1999]



Left Side Right Side

Figure 2-7. Mammograms (Cranio-caudal View). [Smathers, 1999]

2.4.3 Mammography Accuracy.

Even with two viewpoints which maximize the imaging of prospective locations of cancer, mammograms can fail to capture up to 10% of all breast cancers [Olivotto et al, 1995:42]. This figure is different than the previously mentioned 15% false positive rate, which is a mammogram misdiagnosis percentage. The failure of a mammogram to capture a tumor may be because the tumor may be hidden behind other breast tissue or be located in an unusual place within the breast [42]. Also, due to the narrow exposure latitude of mammography, the breast may not be properly compressed when situated by the mammography technician [Harris et al, 1991:90]. The breast must be sufficiently compressed between two rigid plates in order to reduce thickness variations across the breast [90]. Failing to do so results in an image on the mammogram that could be mistakenly read as a "dense, glandular" breast with no visual abnormalities when a correct visualization would show an abnormality [AMWA, 1996:63]. Also, the breast must be properly positioned in order to view as much of the breast tissue as possible. Figure 2-7 is actually an example of when a breast has not been properly positioned within the rigid positioning plates.

2.5 Existing Mathematical Tumor Models.

Interest in tumor propagation and the unsolved mysteries of cancer have spurred the use of mathematics for the purpose of accurately depicting the behavior of tumors, both benign and malignant. Created mathematical models have offered insight into the various growth phases that tumors experience, the interactions between the immune system and tumor cells, and cell population theories. The weaknesses of these models include the

assumptions that simplify the nature of the process being modeled. Current models may assume an ideally spherical tumor, disregard the element of time, set a constant cell cycle growth rate, or use any other assumption that simplifies the modeling procedure.

Several tumor models follow the changes that tumors go through based upon the prevascular/vascular growth phases. In the prevascular growth phase, the tumor is basically spheroid in shape, and grows rapidly to a dormant state [Adam and Bellomo, 1997:xv]. In the next phase, rapid growth reinitiates as nutrients flood the tumor, but begins to slow again as the tumor becomes larger [xv]. Overall tumor growth has been shown to follow a Gompertzian curve, where the growth constant is exponentially slowing [Harris et al, 1991:168]. In other words, the more cells that must divide, the longer the time its takes to accomplish the process in the long run.

Though they offer insight into many important aspects of tumor growth, most current mathematical tumor models offer little to the visual aspects of the tumor structure. However, some tumor models are compared against an *ideal tumor model.* The ideal tumor model begins with one malignant cell and assumes away many of the complicating intricacies of tumor growth. The model assumes ideal growth without cell necrosis, and a constant population doubling rate. In other words, each malignant cell divides and survives. Also, instead of a overall slowing growth rate that follows a Gompertzian curve, the time between population doublings in the ideal tumor model is estimated to be about 100 days [AMWA, 1996:146]. This time period projects a palpable tumor to be formed in about 10 years, while a tumor detected by mammogram would be formed in about 8 years [146]. A depiction of the ideal tumor model can be seen in Figure 2-8. From the diagram, it is obvious that the

probability for a tumor to metastasize increases with time and tumor size, but can occur at virtually any point within the life cycle of the tumor.

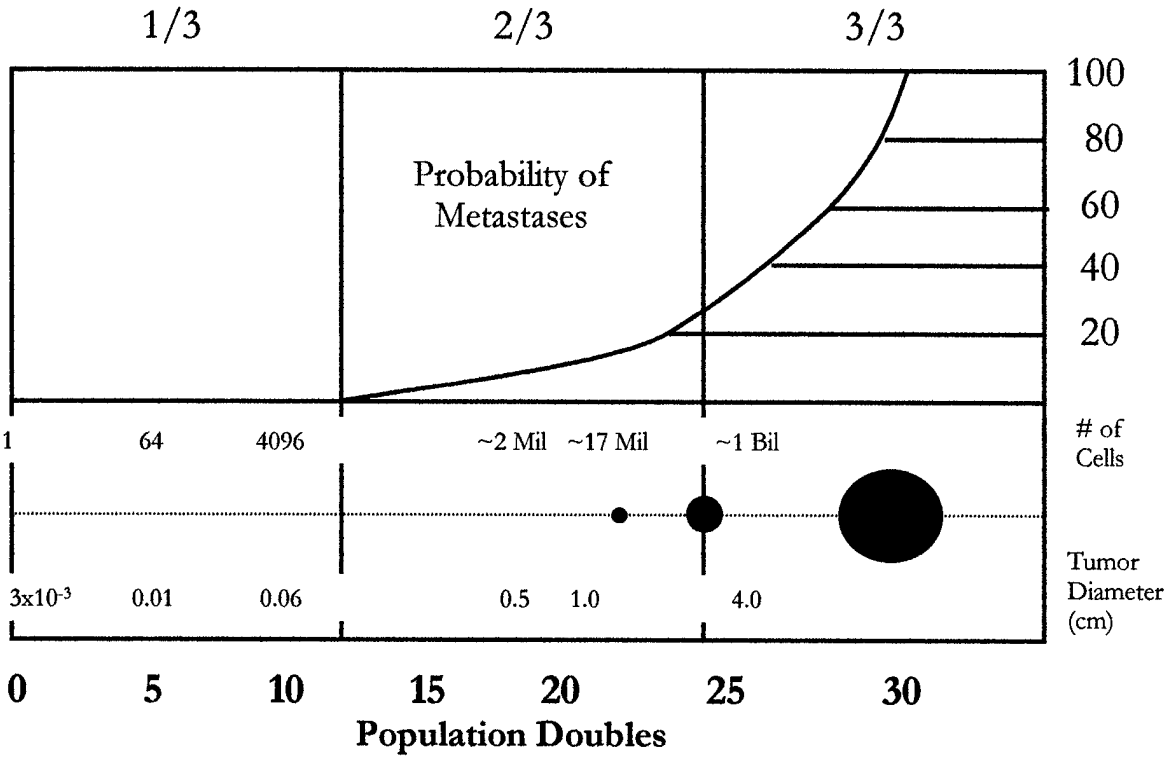


Figure 2-8. Ideal Tumor Model Diagram. [Zander, 1985]

Thus, current math models do not greatly impact the visualization aspects of mathematical tumor model growth, but the ideal tumor model concept is a measuring stick that all model outputs can be compared against. The model provides basic milestones that each breast cancer tumor may attain.

III. Methodology.

3.1 Research Approach.

Since this research is the starting point in a long-term research effort, a definite stepwise pattern to the methodology to solve this problem is taken. The steps designed to provide a pattern of research are:

1. Grasp a basic understanding of how cancer propagates.
    a. Literature review of pertinent material
    b. Interviews with medical expert contacts
    c. Internet searches for applicable material
2. Conduct a tumor information-gathering effort, including:
    a. Cell and tumor sizes
    b. Tumor growth rates in time and probability
    c. Metastasizing probabilities of breast cancer
    d. Breast area distribution of cancer
    e. Any other pertinent data necessary
3. Build a 2-dimensional tumor model:
    a. Cellular level
    b. With simplifying assumptions
    c. Acts in accordance with a "slice" of a tumor
4. Embellish existing 2-dimensional tumor model
    a. Changing random effects
    b. Surrounding breast tissue

c. Simulate invasive cancer

d. Propagates according to theory

5. Extend the model as a 3-dimensional tumor model

    a. Cellular level

    b. Density aspects from different perspectives

6. Embellish 3-dimensional tumor model

    a. View 3-dimensional tumor from many perspectives

    b. Create 3-dimensional breast tissue structure to encounter tumor

    c. Simulate invasive breast cancer

    d. Simulate growth rates associated with growth phases

7. Embellish the model by applying physical forces associated with mammography, if necessary.

8. Create accurate mammogram "image" of X-rays passing through tumor and breast tissue.

9. Perform sensitivity analyses.

    a. Tumor shapes and sizes

    b. Tumor growth rates

    c. Tumor "image" densities

10. Publish a report on the applicable data, trends, and information found.

3.2 Research Tools.

Materials used in this research effort were found at the Fordham Medical Library on the Wright State University campus, the Wright State Library, and the Air Force Institute of Technology (AFIT) Library. The Internet also produced very good informational and visual material on breast cancer and mammography.

3.3 Modeling Tools.

All of the models used in this thesis were programmed in Matlab 5.0 for Windows. The use of this mathematical package offers a tremendous strength in matrix computations and graphics. If follow-on researchers have the knowledge or the inclination, JAVA may be incorporated into producing an applet that runs on the AFIT website and offers users a chance to interact with the model. Also, JAVA may offer a 3-dimensional visualization technique that may be superior to the results of this thesis. Finally, due to the shortcomings of MATLAB's 3-dimensional imaging capabilities, a better technique for producing 3-dimensional tumor representations should be researched further. There appears to be promise with the Tcl/Tk visualization language found within The Visualization Toolkit text [Schroeder et al, 1996:331]. Visualization concepts such as the *Marching Cubes* algorithm, which represent volumes with connected triangles, could provide a better way of looking at the tumor from multiple angles [148].

IV. Model Descriptions and Results.

4.1 Overview.

The modeling process is an evolution of ideas as new problems take the place of old ones. Every step that is taken to make the model more realistic adds more complication to the methodology. Therefore, the tumor model begins at a basic starting place and builds on the realism of the simulation.

Though each model created during this research effort has its own individual characteristics or purpose, there are similar aspects that they all share. Each tumor model establishes a 2-dimensional grid or 3-dimensional set of cells that represents the space in the breast where all of the tumor growth will take place, or *growth space.* No 3-dimensional model allows tumor growth beyond the growth space. This growth space is also analogous to the areas of the breast that would be viewed under a mammogram.

Each model requires input from the user in order to simulate tumor growth. Each model requires the size of the growth space and the number of iterations, or population doubles, that the tumor will undergo to be entered. The 3-dimensional models also allow the user to specify the [x,y,z] coordinates of where the original malignant cell is placed within the growth space.

All models also use a similar technique to differentiate between cells found within the growth space. Since each "cell" within the matrix represents a cell or cell space in the breast, they are a given a corresponding numeric value for the tissue that they represent. For simplicity in explaining model algorithms, these numeric values have been associated with a color. All malignant cells are deemed RED. A YELLOW cell represents either a cell space

to where a malignant daughter cell may divide (Tumor4) or a malignant daughter cell. BLUE represents all empty matrix cells and can be considered as surrounding fat, connective tissue, empty space, or an area where a duct or lobule has been penetrated. BLACK represents the cells that constitute breast structures like the ducts or lobules (3-dimensional models only). Table 4-1 lists the definition of each cell color designation and to which models they apply.

Table 4-1. Cell Color Designation and Model Applicability.

Color Designation	Definition	Used in 2-D Models?	Used in 3-D Models?
RED	Malignant Cell	Yes	Yes
YELLOW	Possible Cell Space for Malignant Daughter Cell	Tumor4	No
YELLOW	Malignant Daughter Cell	Tumor5,6 & 7	Yes
BLUE	Empty Space/Fat/Connective Tissue/Ductal or Lobular Hole	Yes	Yes
BLACK	Breast Tissue (Duct or Lobule)	No	Yes

4.2 2-Dimensional Models.

The first step in the 2-dimensional tumor modeling process is to create a 2-dimensional representation of a malignant tumor. The concept is to have the modeled tumor "slice" attempt to grow and spread on the cellular level like a malignant tumor may if given no boundaries. The following models construct breast structures with which the growing tumor must interact.

4.2.1 2-Dimensional Model Overview.

Four 2-dimensional models exist: Tumor4, Tumor5, Tumor6, and Tumor7. The first, Tumor4, is a successful compilation of subroutines from failed MATLAB programs and ideas from a JAVA tumor growth program. Tumor5 improves upon the tumor growth aspects of Tumor4 by adopting an improved growth algorithm which produced more realistic results and executed much faster. Tumor6 creates impenetrable 2-dimensional boundaries on the image grid. These boundaries represent breast structures, such as ductal walls, that the tumor may encounter during its growth cycle. Tumor7 continues the modeling process by allowing the accumulating tumor mass to pass through the 2-dimensional boundaries, or ductal "slices". For all of the 2-dimensional models, the original cancer cell begins in the center of the 2-dimensional grid.

4.2.2 Tumor4.

The first model begins with the construction of a square, 2-dimensional grid of a size specified by the user. The center cell is then designated as the malignant cell. The malignant cell is then ushered into the growth algorithm where it doubles as many times as the user has specified. The Tumor4 model is the only model which uses the original tumor growth pattern conceived at the beginning of the thesis work. A visual demonstration of the Tumor4 algorithm can be seen in Figure 4-1. The growth algorithm works in the following manner:

1. Existing RED cell(s) designate the surrounding BLUE cells as YELLOW. If the surrounding cell is already YELLOW, then the probability for the cell to become RED is increased. For each RED cell, there should exist eight YELLOW cells unless the surrounding area is already occupied by a RED cell.

2. The YELLOW cells are assigned a probability of occurrence based on the total value of YELLOWs. These probabilities are then summed into a cumulative probability distribution.
3. Random numbers associated with the appropriate number of dividing RED cells are generated, i.e. if two RED cells exist, then two random numbers must be generated.
4. A YELLOW cell is selected to become RED when a random number is less than the cumulative probability distribution associated with the appropriate cell.
5. Return to step 1 if more iterations (population doubles).

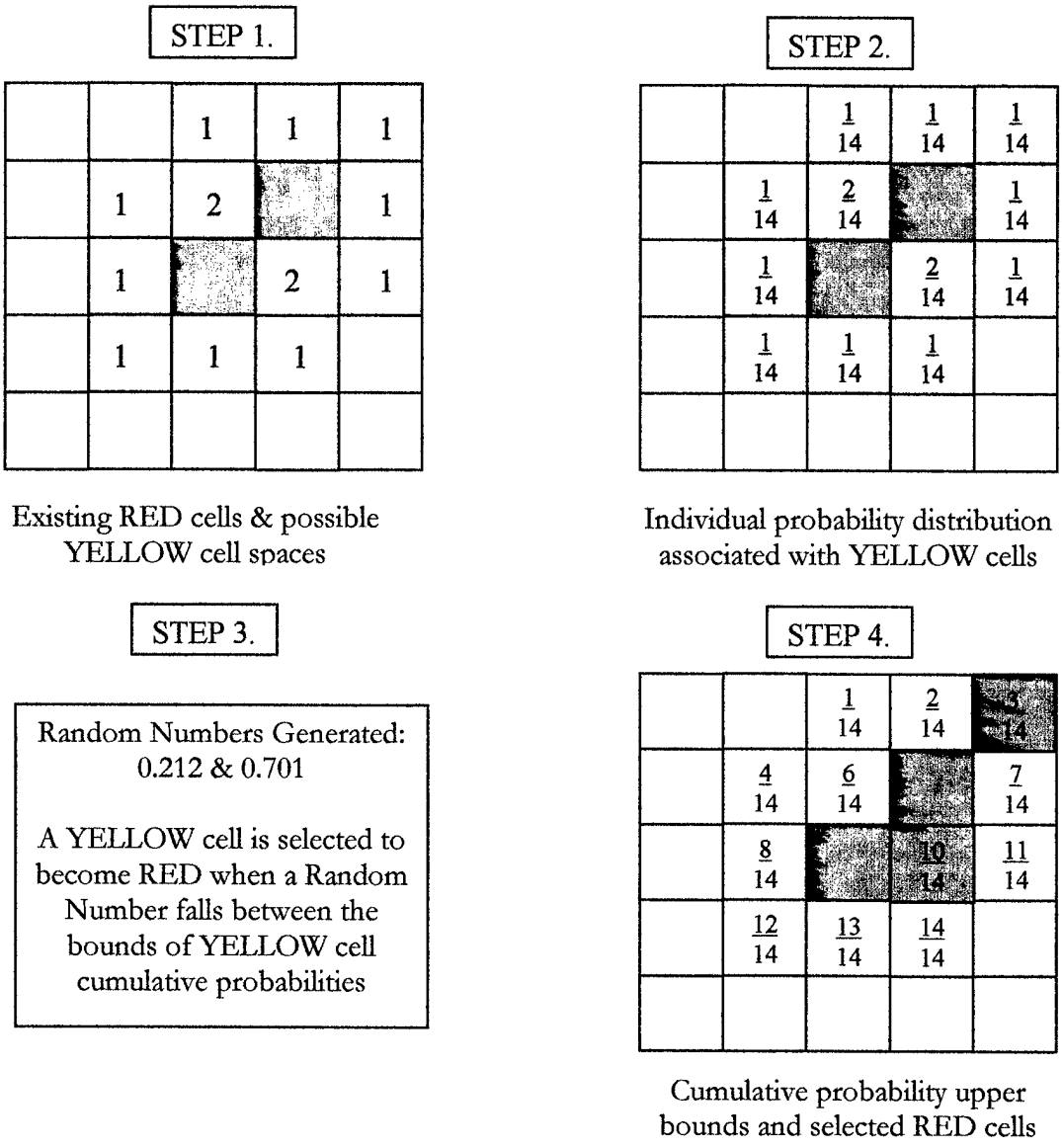


Figure 4-1. Tumor4 Growth Algorithm.

For small population growths the Tumor4 model is adequate. However, Tumor4 is not ideal in two regards. First, the model takes a long time to execute. Increasing the population doubles increases the amount of time it takes to complete the algorithm exponentially. More importantly, it is limited in that as the tumor grows larger, the number

of RED cells that divide soon outnumber the possible YELLOW cells. The resulting tumor takes on a block-like appearance as the YELLOW space begins to become too small. This phenomenon occurs around eight population doubles. Figure 4-2 is a collection of consecutive images generated by the Tumor4 model. The tumor model pictured generated 2^8 cells across a 37 x 37 square grid. The Tumor4 model allows the user to see the generation growth of the tumor by associating a different color with each population doubling.

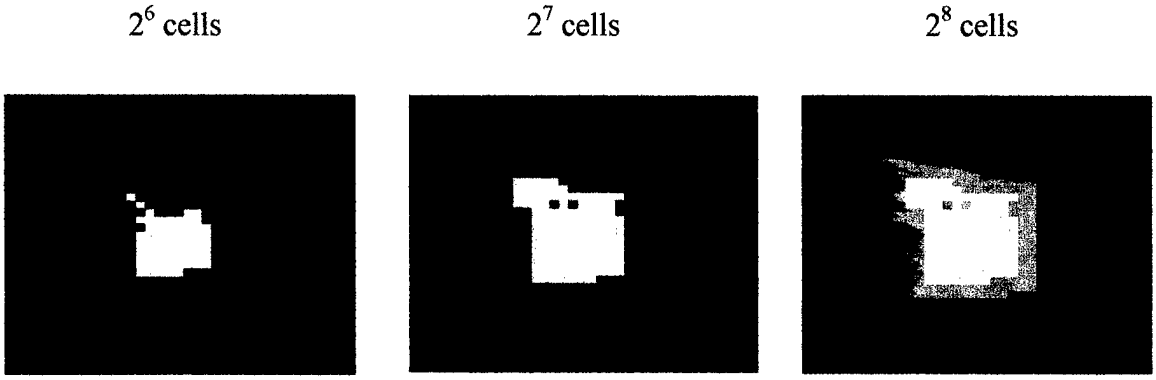
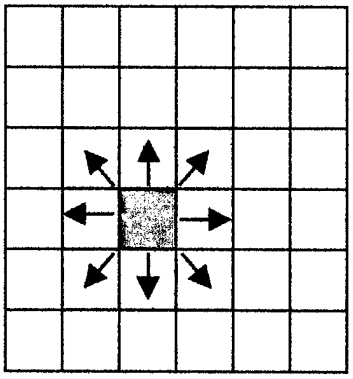


Figure 4-2. Tumor4 Sample Output.

4.2.3 Tumor5.

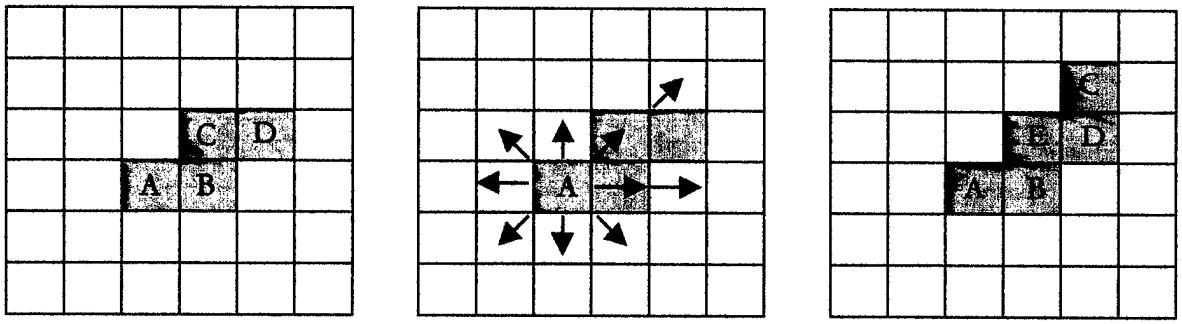
The Tumor5 model improves upon the Tumor4 model by incorporating a new tumor growth algorithm. Instead of choosing the possible cells that a malignant cell can move to, like Tumor4, the new algorithm allows each dividing RED cell to choose from one of eight directions in which the malignant daughter cell will travel (Figure 4-3a.). So, instead of assigning the probability of occurrence to the surrounding cell spaces, the Tumor5 model places the probability that a cell space will be occupied by a malignant cell on the *direction* in

which the dividing cell will travel. The daughter cell of the dividing RED cell, represented by a YELLOW cell, travels in the selected direction until it reaches a BLUE cell. The daughter cell continues in the given direction if it encounters an existing RED cell or an existing YELLOW cell (Figure 4-3b.). This algorithm can be thought of as a "pushing" algorithm. The daughter cell "pushes" the existing RED and YELLOW cells to the empty region as it divides.



Eight Possible Directions for
Daughter Cell to Travel

Figure 4-3a. Tumor5 Growth Algorithm.



Existing RED cells

Cell "A" Divides--
Daughter Cell [E] Has
Eight Options

Resulting RED Cells

Figure 4-3b. Tumor5 "Pushing" Technique.

The Tumor5 model is a better model than Tumor4 because it represents an actual tumor growth sequence more accurately, takes less time to execute, and is inherently simpler. Figure 4-4 shows a sample image from the output of the Tumor5 MATLAB model. The tumor "slice" pictured contains 2^8 cells over a 37 x 37 square grid. Notice the more rounded shape when compared with the tumor images from theTumor4 model in Figure 4-2.

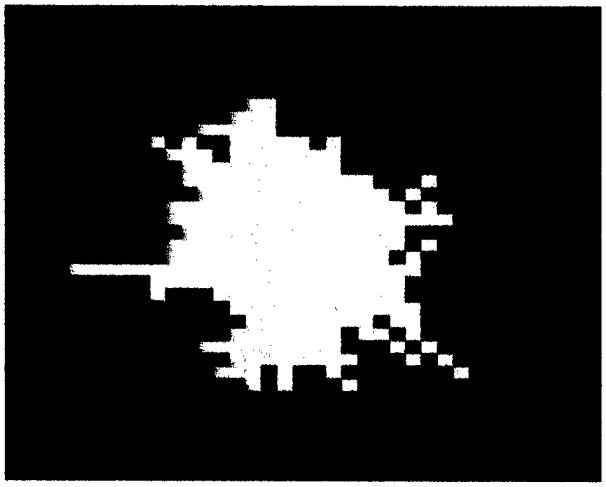


Figure 4-4. Tumor5 Sample Output.

4.2.4 Tumor6.

Tumor6 builds on the Tumor5 algorithm by incorporating impenetrable boundaries, represented by BLACK cells, within the tumor growth space. This model is merely a stepping stone to reaching the next step: a penetrable boundary that represents an actual female breast duct or lobe. An example of the tumor growing against the boundary can be seen in Figure 4-5. Again, the figure depicts a tumor that has experienced eight population doubles distributed over a 37 x 37 grid. Notice the squishing that has occurred due to the presence of the boundaries.

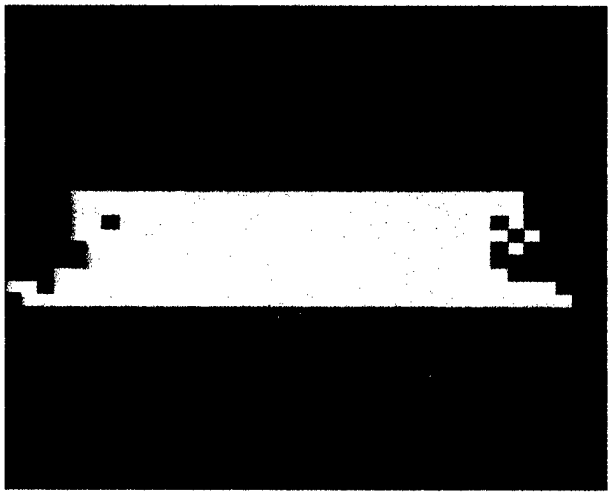
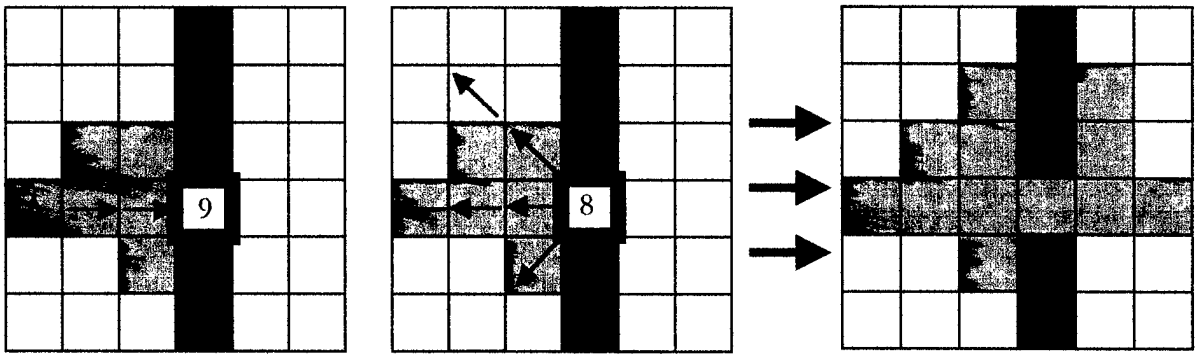


Figure 4-5. Tumor6 Sample Output.

4.2.5 Tumor7.

Tumor7 creates a penetrable boundary which the tumor mass can break through when the cancer has acquired the ability to infiltrate the cells that constitute the boundary layer. This model attempts to produce results that mimic the infiltrating aspects of the theoretical cancer growth sequence (Figure 2-4). To accomplish this, the cells of the 2-dimensional boundaries are given a certain amount of "hit points". For each time that a traveling YELLOW cell encounters a boundary cell, a hit point is removed from the boundary cell and the YELLOW cell must be redirected. When the hit point value of the individual boundary cell reaches a minimum value, the boundary cell no longer exists. Thus, malignant daughter cells may pass through the cell space where the boundary once existed. Figure 4-6 gives an example of this algorithm. The modeled tumor mass breaking out of the duct structure represents anaplasia, or the invasiveness of breast cancer, as it quickly grows and spreads. This phenomenon is illustrated in the image created by the Tumor7 model

(Figure 4-7). Again, the image is of a tumor that has undergone eight population doubles and is spread across a 37 x 37 square grid.



Malignant Cell 'A' Divides and Encounters Boundary Cell with 9 Hit Points

Daughter Cell Must Redirect at Boundary Surface; One Hit Point Deducted From Boundary Cell

Malignant Daughter Cells May Pass Through When Boundary Cell Hit Points Equal Zero

Figure 4-6. 2-Dimensional Tumor-Boundary Interaction Algorithm.

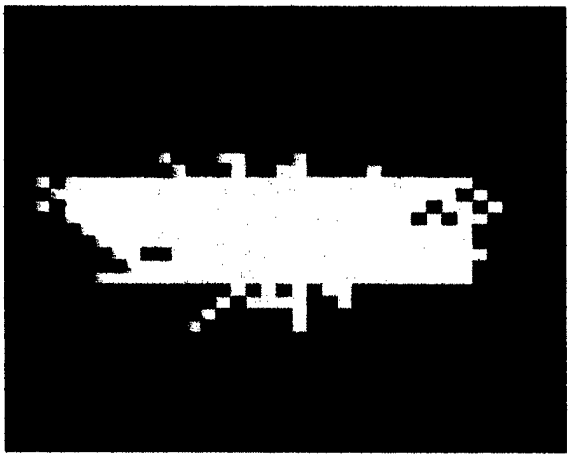


Figure 4-7. Tumor7 Sample Output.

4.3 3-Dimensional Models.

There are certain aspects that must be addressed by a 3-dimensional tumor model representation which simply cannot be appreciated with the created 2-dimensional models. A mammogram relies on tumor mass density to be useful in detecting breast cancer. The mammographic shape of a lesion's density allows examiners to establish insight on position and differentiates between malignancy and structure types. Thus, it is imperative for a model to depict tumor tissue density. Also, the position of a malignant tumor *in vivo* cannot be known prior to a mammogram. Therefore, a model which depicts several viewpoints is also necessary. Finally, the variability of the tumor's growth pattern could be drastically different when realized in three dimensions. Unfortunately, none of the 2-dimensional models created can answer these problems. Thus, 3-dimensional tumor growth models are crucial to the realization of these aspects.

The first step in the 3-dimensional tumor modeling process is to create a 3-dimensional representation of a malignant tumor. The idea is to have a tumor mass model attempt to grow and spread on the cellular level like a malignant tumor may if given no boundaries. The following models simulate breast structures with which the growing tumor must interact.

4.3.1 3-Dimensional Model Overview.

Three 3-dimensional models exist: Tumor3d, Tumor3d2, and Tumor3d3. The first program merely allows the tumor to grow in three directions unhindered by constructed boundaries. The second erects impenetrable walls in three dimensions. Tumor3d3 allows the growing tumor mass to break through the simulated boundary.

Due to limitations in MATLAB, visualizing the tumor in three dimensions is much different than in two dimensions. The growth space in three dimensions is much like a cube. Figure 4-8a illustrates what the 3-dimensional growth space looks like. However, the resulting images in MATLAB are merely the 2-dimensional *faces* of the cube. The perspective from which these 2-dimensional face images are taken are delineated in Figures 4-8b and 4-8c. Another major difference in visualizing the simulation is that the tumor images seen are actually tumor cell densities. The numeric value for each cell is summed across the 2-dimensional face of the cube. The result is an image that looks *through* the tumor's mass. This image is analogous to viewing tumor densities on a mammogram. By examining the different faces of the cube, an idea of how the tumor is growing can be established — a concept not easily ascertained with the use of mammography. Figure 4-9 supplies the axes of reference for the resulting images created by MATLAB for each 3-dimensional tumor model.

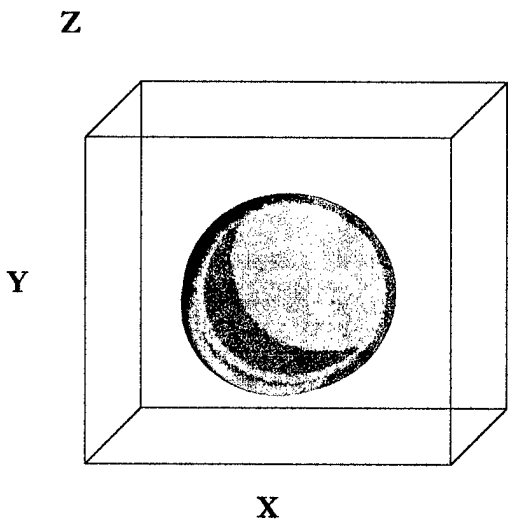


Figure 4-8a. 3-Dimensional Visualization (Tumor3d).

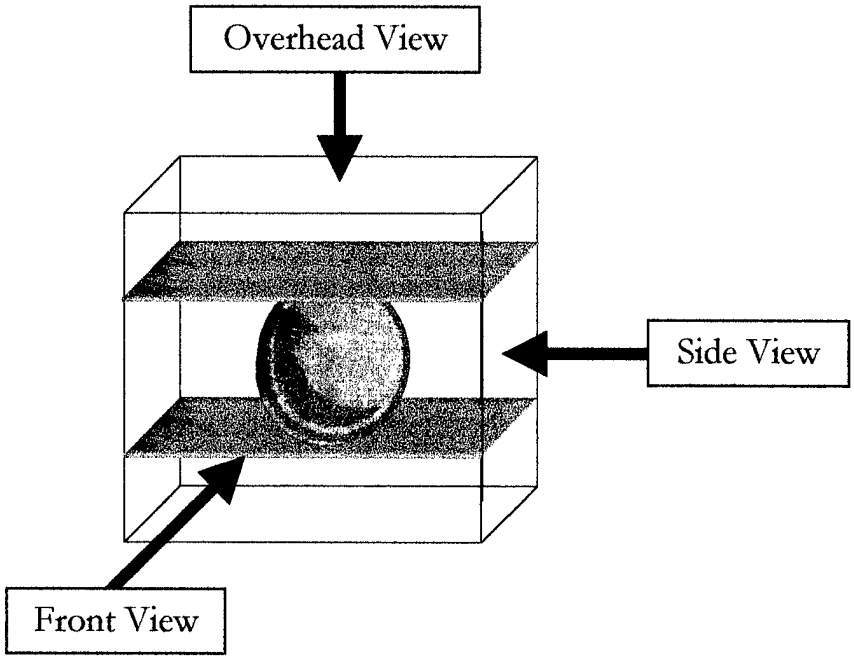


Figure 4-8b. 3-Dimensional Visualization (Tumor3d2).

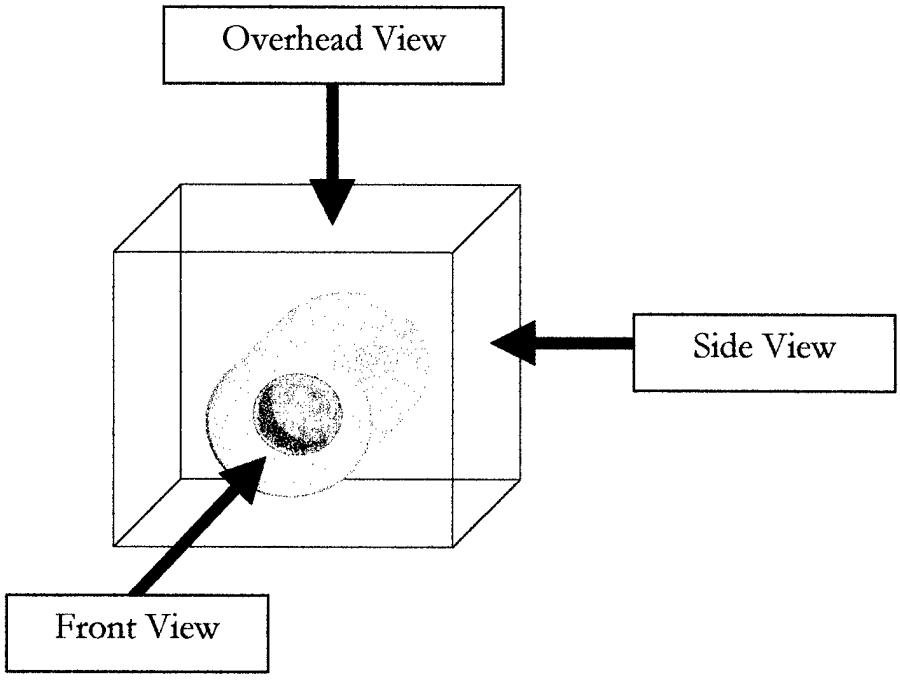


Figure 4-8c. 3-Dimensional Visualization (Tumor3d3).

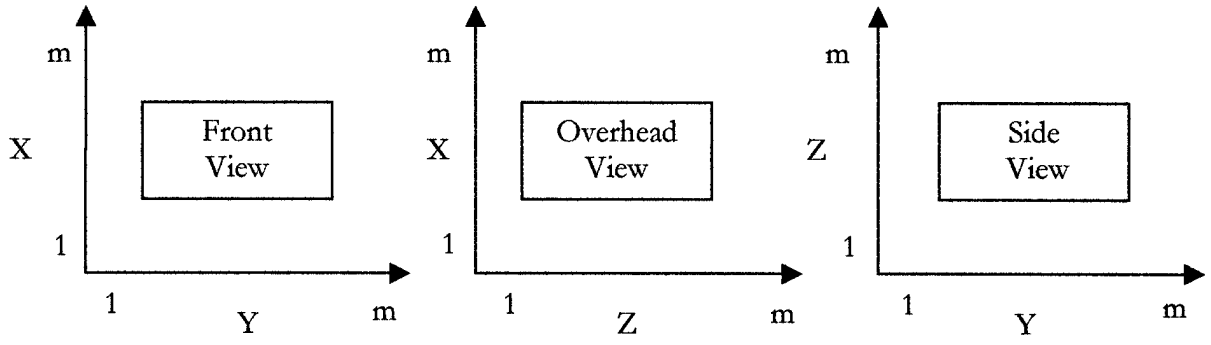


Figure 4-9. 3-Dimensional Image Axes of Reference.

4.3.2 Tumor3d.

Each 3-dimensional tumor model incorporates the same *pushing* logic algorithm used in the latter 2-dimensional models. The difference is that each RED cell now has 26 different directions in three dimensions to choose from rather than the eight possible directions offered in two dimensions. Tumor3d allows the tumor to freely grow within the confines on the growth space. Figure 4-10 offers images created from a run of the Tumor3d model. The generated tumor is a collection of 2^8 cells in a 25 x 25 x 25 growth space volume. Notice that the areas of lighter shading represent a more dense portion of the tumor mass. The original malignant cell began at the location [12,12,12].

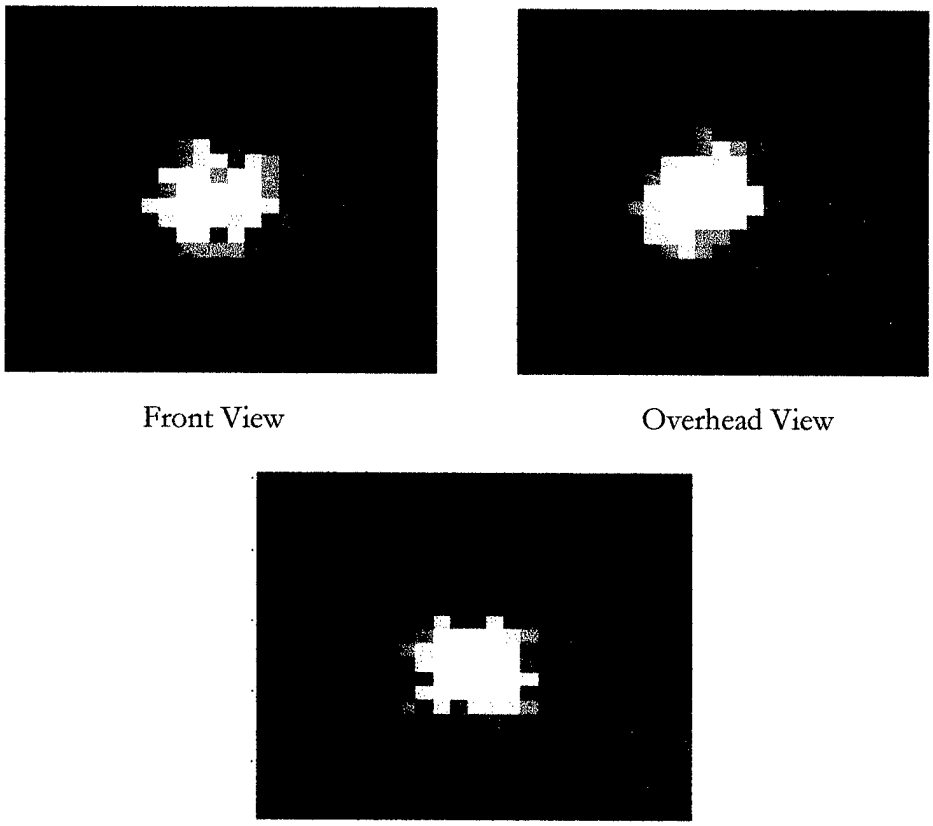
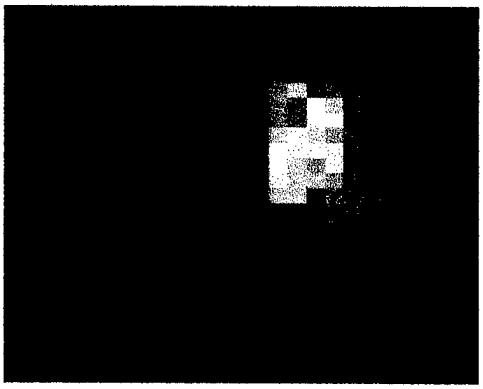


Figure 4-10. Tumor3d Sample Output.

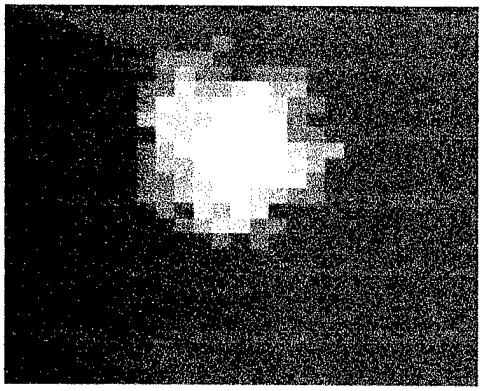
4.3.3 Tumor3d2.

Using the Tumor3d MATLAB code as a basis, the Tumor3d2 model constructs two horizontal planes within the 3-dimensional growth space in the manner presented in Figure 4-8b. These planes are not designed to be breached. Thus, the cells will merely spread within the confines of the walls. The images generated by Tumor3d2 will make the growth space appear to contain vertical walls, but Figure 4-9 illustrates the true orientation of the axes and prove that there are horizontal planes surrounding the tumor. The tumor

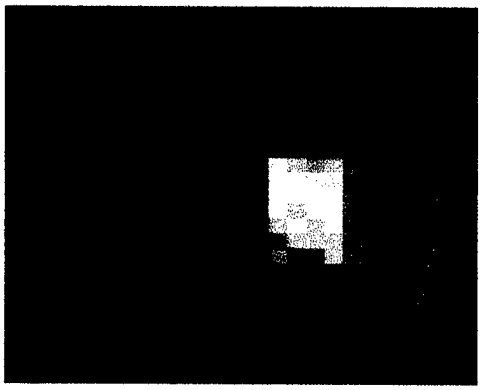
generated from the Tumor3d2 model, which consists of 2^8 cells, is placed in a 25 x 25 x 25 3-dimensional growth space at coordinates [15,15,13], is seen in Figure 4-11. Notice that the color of the Overhead View is much lighter than the other two views. This is due to the fact that the perspective takes into account the density of the walls. So, the density value of both walls is being summed up with the density value of the tumor as the image is being generated.



Front View



Overhead View



Side View

Figure 4-11. Tumor3d2 Sample Output.

4.3.4 Tumor3d3.

The final 3-dimensional model takes advantage of the previous models' successes. In learning from the mistakes of earlier 2-dimensional models, Tumor3d3 also incorporates a subtly different algorithm for tumor interaction with boundary cells. Instead of redirecting the daughter cell at the cell boundary, which can lead to run-time errors with MATLAB, Tumor3d3 has the cell redirect at the cell space right before it encountered the boundary cell. An example of this revised algorithm is depicted in Figure 4-12.

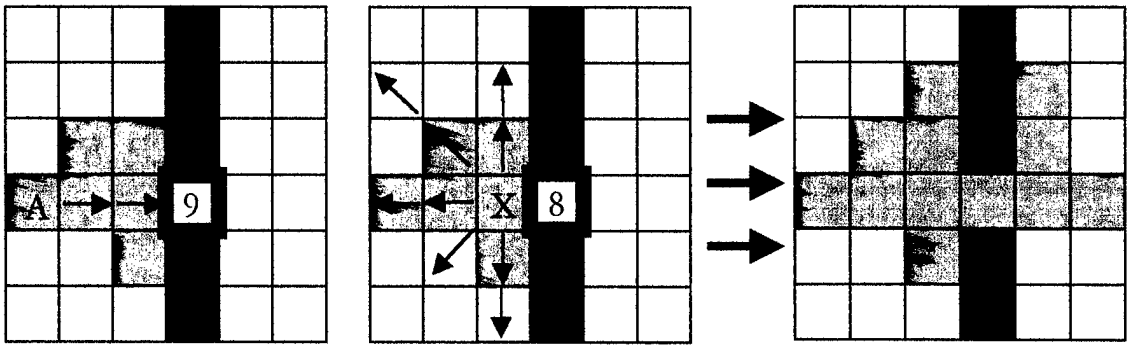
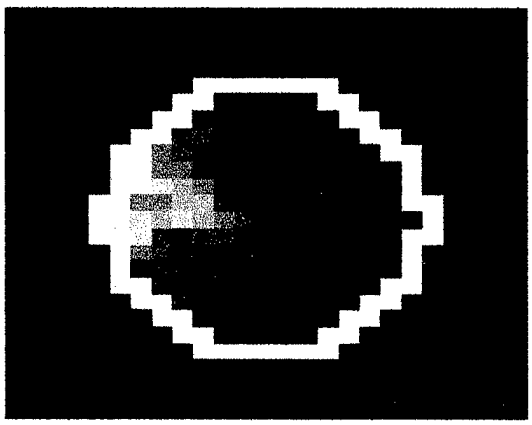


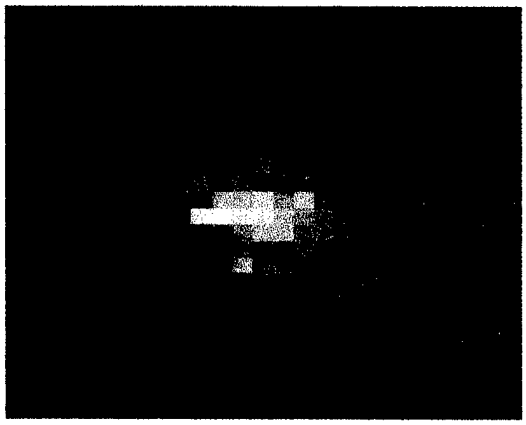
Figure 4-12. 3-Dimensional Tumor-Boundary Interaction Algorithm.

The result of Tumor3d3 can be visualized in Figure 4-13. The generated tumor consists of 2^8 cells distributed within a 25 x 25 x 25 growth space and started at the point (12,7,13). The structure seen within the 3-dimensional cube is a cylinder that represents a duct within the

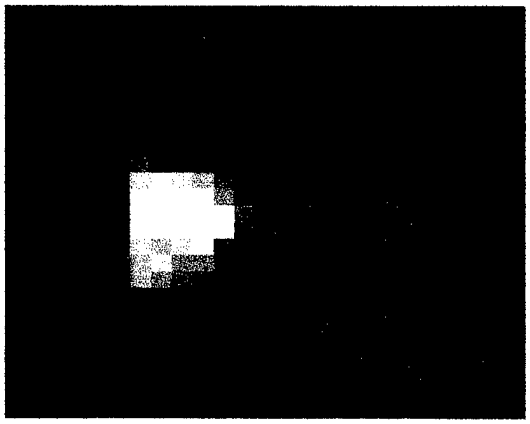
female breast. Notice the lighter colors on all of the views, which represent the density of the cylinder being summed up along the faces of the image. In the Tumor3d3 model, the tumor does have the capability to become invasive and seep through the duct using the algorithm described in Figure 4-12. The tumor depicted in Figure 4-13 is progressing to a state of ductal carcinoma in situ. Again, refer back to Figure 4-9 for the proper viewpoint axes alignment.



Front View



Overhead View



Side View

Figure 4-13. Tumor3d3 Sample Output.

V. Conclusions and Recommendations.

5.1 Introduction.

It is important to remember the following phrase when simulating a real world system: "All models are wrong, but some are useful". For instance, no computer model could ever predict what value a die will show when rolled (at least with present technology in mind!). However, we can model the long-term values shown when rolling the die. With an accurate model, we can then make educated guesses at what percentage of values the dice will show in 1000 rolls, for example. This same concept applies to modeling breast cancer tumors. From the original malignant cell, the created models will never be able to tell in which direction an actual tumor will grow, if it will become invasive, or to which organ it will metastasize. However, if the model has an accurate growth algorithm and utilizes necessary embellishments, then, given certain situations, it could offer insight into the percentage of the time that a tumor may become invasive or the probability it will grow in a certain direction. In fact, by studying the statistical characteristics of the 2-dimensional projections produced by the model, we should be able to ascertain an overall "look" of young tumors.

The models created for this research effort are admittedly wrong, but they could offer insight later as they are improved. The important step now is to understand how well the growth and development algorithms work. In order to see how the models perform, an adequate measuring stick is needed for comparison. The obvious choice is to compare the images created by the MATLAB code to actual mammograms or pictures of tumors found in female patients. This approach lacks applicability in that the model assumes away many of the aspects of an actual tumor. Also, the final model (Tumor3d3) depicts tumor density,

not slices of dissected tumors. However, comparison of the model to the real world should provide a glimpse of the decisions on the modeling road ahead. The next, and only logical, alternative is to compare the models against the theoretical developmental pattern of breast cancer, as seen in Figure 2-6. To be successful, the Tumor3d3 model should at least accurately mimic this notional tumor development process.

5.2 Results.

When compared against the theoretical breast cancer development sequence, the images of the most current tumor model needs to be seen for what they do not include. The Tumor3d3 model does not address the tumor development aspects, such as cell necrosis, growth rates, malignant cell heterogeneity, accurate cell proportionality to breast structures. However, most of the realism lost in the Tumor3d3 model is also absent from the ideal tumor model. In fact, the Tumor3d3 is a algorithmic representation of the ideal tumor model. With this in mind, the Tumor3d3 output should be close to what the theoretical breast cancer development sequence depicts. Figure 5-1 shows the theoretical breast cancer development sequence next to a series of MATLAB images produced by the Tumor3d3 model. Notice that the cell growth patterns are remarkably similar. This comparison clearly illustrates how well the Tumor3d3 model depicts theoretical tumor growth, especially within the confines of the breast duct structure. Figure 5-2 is a picture of an actual invasive carcinoma cross-section next to output from the Tumor3d3 model (2^{12} malignant cells in a 25 x 25 x 25 growth space where the boundary cells have 7 hit points). These images show how much detail is lost when the model is compared to the real world problem and indicates the work ahead.

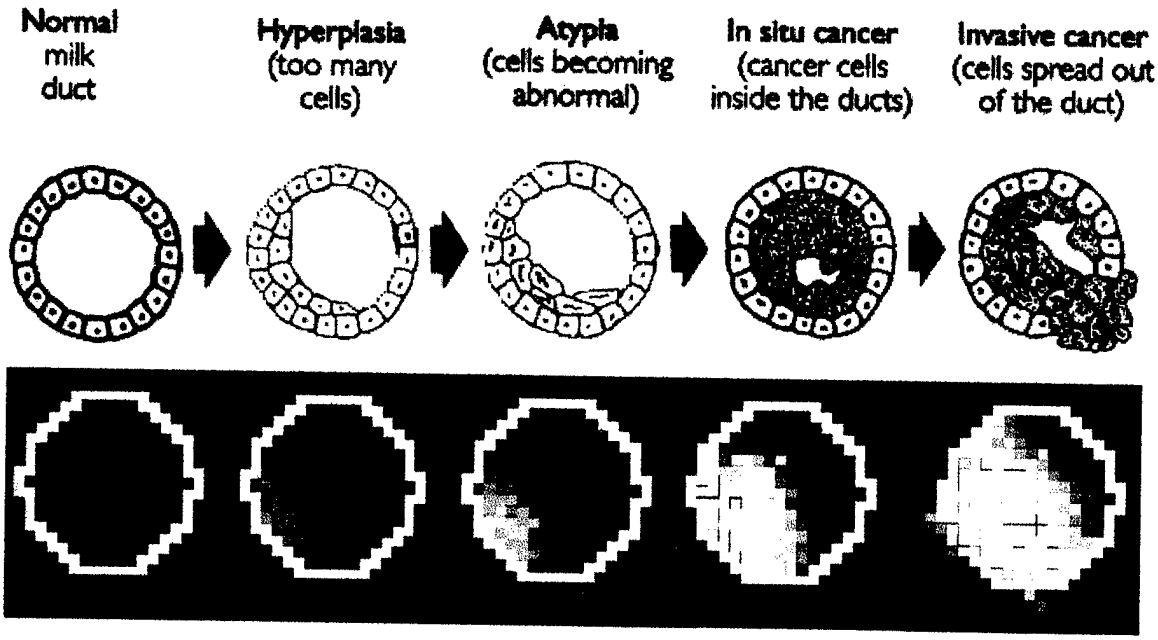


Figure 5-1. Tumor3d3 Comparison to Theoretical Tumor Development Sequence.

[Olivotto, 1995]

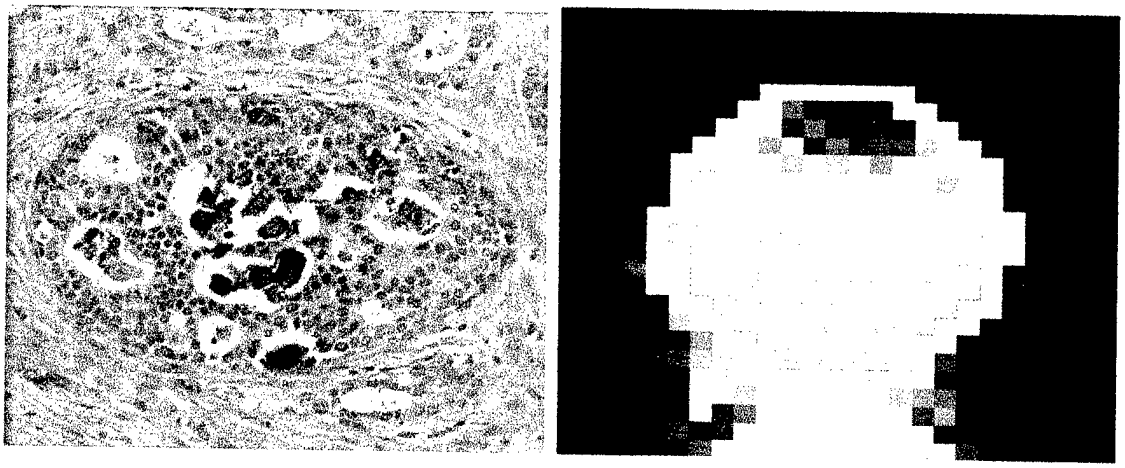


Figure 5-2. Tumor3d3 Comparison to Invasive Ductal Carcinoma. [Klatt, 1999]

5.3 Conclusions.

From these comparisons, the conclusion is that the Tumor3d3 is a good foundation model on which future models can be based. The model presents tumor growth that acceptably mimics the growth sequence that breast cancer is thought to follow. Also, the model represents the ideal tumor growth model in algorithmic form. Therefore, the 3-dimensional models presented are an excellent starting point for further modeling. Overall, the 2-dimensional models are basically learning steps on the way to the realization of the 3-dimensional models, which better handle the problems of tumor growth. Though the tumor models generate encouraging images, the results from Figure 5-2 point to the realism that must be captured in future models.

5.4 Recommendations.

Since the solution to the problems associated with modeling real world breast cancer tumors will require further work and research, there are several aspects of the modeling approach that can be embellished by follow-on research efforts. These aspects can be divided into three different categories: improvements on the tumor model by including additional tumor characteristics; embellishments to the visualization of the model by adding the physics of tissue manipulations to x-rays, and improvements to the 3-dimensional visualization aspects of the tumor models.

Possible improvements to the current tumor model can be attempted through the incorporation of additional tumor characteristics. For instance, as the tumor model grows, a certain percentage of cells could be programmed to die out at each population doubling. This corresponds to the concept of cell necrosis and could greatly impact the shape of the

tumor or the density of the resulting image. Additionally, an embellishment to the model could be made to incorporate cell heterogeneity. This could be accomplished by allowing malignant RED cells to hold varying numeric values that correspond to a different density. Finally, the MATLAB model could be changed to incorporate multiple, adjustable breast structures, varying cell growth rates, and varying growth conditions whether the tumor is inside or outside the duct or lobule.

In addition to the model embellishments, a MATLAB graphical user interface (GUI) could be created that allows the user to input the required initializing information for creating a realistic tumor growth space and a tumor with variable growth dynamics. A rough concept for this GUI is illustrated in Figure 5-3. It is also recommended that all models be executed in the professional version of MATLAB. This would allow larger growth spaces and should cause the model to execute more quickly.

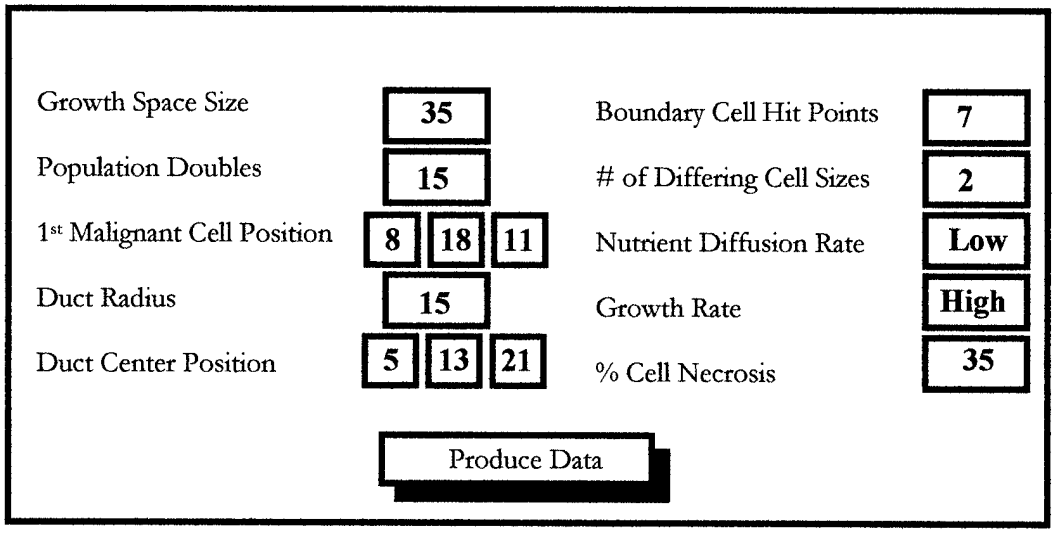


Figure 5-3. Possible MATLAB GUI.

The visualization elements of the tumor model should correspond directly with that of a mammographic environment. Thus, knowing how the tumor and surrounding breast structure tissue reacts with x-rays from a detection source is crucial to capturing a realistic image of the modeled tumor density. Therefore, research into the mechanics of these processes is recommended.

Akin to the mammographic aspect of the modeled tumor, the 3-dimensional visualization should also be as realistic as possible. Since the tumor could be in any area of the breast and at any angle, the model's visualization options should not be limited to the 3 viewpoints offered by the MATLAB output. The model should offer the ability to rotate the generated tumor in any plane and capture the resulting 2-dimensional image from any direction. This could be accomplished through the use of an outside visualization software packet. The MATLAB model could generate a set of cell data instead of an image. This data could be sent to the visualization package where it is processed and transformed into a 3-dimensional surface. One such algorithm for create such a surface is the *Marching Cubes* algorithm described in The Visualization Toolkit text [Schroeder et al, 1996:146-154].

These embellishments, when applied to the overall methodology presented, should construct a model that accurately represents a breast cancer tumor. Hopefully, for the sake of those afflicted with the disease of breast cancer, the final model will reach the goals established by this effort.

Appendix A – MATLAB Code for 2-Dimensional Models

There are four 2-dimensional models created for use by this thesis effort. The following table lists which subroutines are executed with each model, what each model accomplishes, and what command runs the model. It is important to note that when running these models, MATLAB will return an error if the MATLAB directory is not pointed at the directory that contains all of the subroutines that are associated with the model to be executed. Also, it is imperative to understand the limitations of each MATLAB package. For example, the Student Version of MATLAB (Version 5.0.0.4073) only allows a 127 x 127 grid to be created in two dimensions. Also, any population doubles (iterations) greater than 12 may overwhelm the computer memory. The 2-dimensional models that create boundaries are also very volatile. The algorithm used for tumor-boundary interactions is faulty (improved in the 3-dimensional version).

Following Table A-1 is the MATLAB code for each model. Comments begin with a "%". Any commented MATLAB commands, i.e. not sentences, are merely commands that were used to check for errors or changes in code.

Table A-1. 2-Dimensional Model Information.

Model	Associated Subroutines	Execution Command	Model Output
Tumor4	ASSIGNLEFT.m ASSIGNRIGHT.m CHOOSE4.m CHOOSECELL.m FINDBORDER.m NUMYELLOW.m RESET.m TOTALREDS.m Tumor.m YELLOWDIST.m	>>*Tumor* Enter an odd number of rows for grid. >> *Size* Enter the number of iterations to be performed. >> *Cells*	A *Size* x *Size* image that contains 2^{Cells} cells within it. The first cell is situated in the center of the grid. Tumor4 uses the first algorithm, which is faulty and time-consuming.
Tumor5	ASSIGNLEFT.m ASSIGNRIGHT.m CHECKDIR.m CHOOSE5.m DIRECTION.m DIRECTION2.m SETRED.m TOTALREDS.m Tumor5.m	>>*Tumor5* Enter an odd number of rows for grid. >> *Size* Enter the number of iterations to be performed. >> *Cells*	A *Size* x *Size* image that contains 2^{Cells} cells within it. The first cell is situated in the center of the grid. Tumor5 uses the improved algorithm.
Tumor6	ASSIGNLEFT.m ASSIGNRIGHT.m BOUNDARY.m CHECKDIR.m CHOOSE6.m DIRECTION.m	>>*Tumor6(Size, Cells)*	A *Size* x *Size* image that contains 2^{Cells} cells within it. The first cell is situated in the

Tumor6 (cont.)	DIRECTION2.m SETRED.m TOTALREDS.m Tumor6.m		center of the grid. There are also impenetrable horizontal lines in the image.
Tumor7	ASSIGNLEFT.m ASSIGNRIGHT.m BOUNDARY.m CHECKDIR.m CHOOSE7.m DIRECTION.m DIRECTION2.m SETRED.m TOTALREDS.m Tumor7.m	*>>Tumor7(Size, Cells)*	A *Size* x *Size* image that contains 2^{Cells} cells within it. The first cell is situated in the center of the grid. There are also penetrable horizontal lines in the image.

MATLAB Code for Tumor4

ASSIGNLEFT.m

```
if LEFT == 1
   LEFT = 1;
else
   LEFT = LEFT - 1;
end
```

ASSIGNRIGHT.m

```
if RIGHT == SIZE
   RIGHT = SIZE;
else
   RIGHT = RIGHT + 1;
End
```

CHOOSE4.m

```
% Initialized and reset variables
POSSIBLE = 0;     LAST = 0;
```

```
SETTING = 0;          ABS1 = 0;
ABS2 = 0;             NUMREDS = 0;
PROB = 0;             RANDOM = 0;
%Created routines that accomplish the necessary tasks
FINDBORDER;
NUMYELLOW;
YELLOWDIST;
%Each RED cell will multiply for a number of generations (user's
   input, CELLS)
%A new random number is selected each pass
%New Area to examine is computed each time through absolute
   distances from CENTER
for k = 1:NUMREDS
   CHOOSECELL;
end
RESET;
%Produce visual image of matrix.  The uint8 command puts matrix %values
between 0 and 255.  This shows color variation for our  %purposes.
image(uint8(A));
% Change area to check for cells
if ABSMAX > TICKER
   ASSIGNLEFT;
   ASSIGNRIGHT;
   TICKER = TICKER+1;
end
PASS=PASS+1;
TOTALREDS;
%NUMREDS
%EXPECTED = 2^PASS
DIFF = (2^PASS) - NUMREDS;
%DIFF
while DIFF > 0
   ASSIGNLEFT;
   ASSIGNRIGHT;
   TICKER = TICKER + 1;
   NUMREDS = 0;
   SETTING = 0;
   FINDBORDER;
   NUMYELLOW;
   YELLOWDIST;
   for k=1:DIFF
      CHOOSECELL;
   end
   TOTALREDS;
   DIFF = (2^PASS) - NUMREDS;
  % NUMREDS
  % EXPECTED
  % DIFF
end
%Produce visual image of matrix
image(uint8(A));
CHANGE = CHANGE+PASS;
```

CHOOSECELL.m

```
RANDOM = rand;
  for i = LEFT:RIGHT
     for j = LEFT:RIGHT
        if A(i,j)< RED & A(i,j)~=BLUE & A(i,j)>RANDOM &
                  SETTING==0
           A(i,j)= RED + CHANGE;
           SETTING = 1;
           ABS1 = abs(i-CENTER);
           ABS2 = abs(j-CENTER);
           if ABS1 > ABS2
              ABSOLUTE = ABS1;
           else
              ABSOLUTE = ABS2;
           end
        end
     end
  end
  SETTING = 0;
  if ABSOLUTE > ABSMAX
     ABSMAX = ABSOLUTE;
  end
```

FINDBORDER.m

```
% Check Area around RED cells.
% If BLUE, then make them YELLOW, even if they are already
% YELLOW
% Also, count the number of RED cells
for i = LEFT:RIGHT
   for j = LEFT:RIGHT
      if A(i,j) >= RED
         NUMREDS = NUMREDS + 1;
         for k = -1:1
            for l = -1:1
               if A(i+k,j+l)<=RED
                  A(i+k,j+l)=A(i+k,j+l)+YELLOW;
               end
            end
         end
      end
   end
end
```

NUMYELLOW.m

```
%Count value of YELLOW cells for probability calculations
for i = LEFT:RIGHT
   for j = LEFT:RIGHT
      if A(i,j)<=RED & A(i,j)~=BLUE
         POSSIBLE = POSSIBLE + A(i,j);
      end
   end
```

```
end
PROB = 1/POSSIBLE;
POSSIBLE = 0;
```

RESET.m

```
%The RESET routine sets all YELLOW cells to BLUE in order to
% calculate the total number of RED cells more easily
for i = LEFT:RIGHT
   for j = LEFT:RIGHT
      if A(i,j)<RED
         A(i,j)=BLUE;
      end
   end
end
```

TOTALREDS.m

```
NUMREDS = 0;
for i = LEFT:RIGHT
   for j = LEFT:RIGHT
      if A(i,j) >= RED
         NUMREDS = NUMREDS + 1;
      end
   end
end
```

Tumor.m

```
RED = 25;            %Sets the color value of the Main matrix (A)
BLUE = 0;            %Equivalent to the background color (empty
                        cell)
YELLOW = 1;          %Used in calculating probabilities for cell
                        propagation
POSSIBLE = 0;        %Sum of possibilities in computing
                        probabilities
LAST = 0;            %Tracks last value in cumulative probability
SETTING = 0;         %When = 1, means a new cell has been
                        selected
LEFT = 0;            %Represents lower index of inner matrix
RIGHT = 0;           %Represents upper index of inner matrix
TICKER = 0;          %Distance of inner matrix from CENTER cell
ABSOLUTE = 0;        %ABS1, ABS2, ABSOLUTE, & ABSMAX are used to
                        determine
ABS1 = 0;            %   changes in inner matrix
ABS2 = 0;
ABSMAX = 0;
PASS=0;              %Number of generations completed
COUNT = 0;           %Used in counting number of RED cells at end
                        of simulation
CHANGE = 1;          %Changes color of cells to show generation
                        change

%   The section of cells that represents the tumor and the
%     surrounding normal cells is represented by the A matrix.
```

```
%   The main section of Tumor.m file allows the user to input
%      1.) the size of the grid to place the A matrix on (given
           by SIZE)
%      2.) the number of generations to grow the tumor cells
           (given by CELLS)
%   The A matrix is initialized as a matrix of zeros and then
           the center cell is chosen to be carcinogenic and is
           allowed to spread. The term RED indicates a malignant
           cell, while BLUE is an empty cell.  The term YELLOW
             indicates the possible places on the grid where the
           tumor cells can move to.  The LEFT and RIGHT values
           keep track of an inner matrix that gives a closer
           boundary to the cancer growth area.  This
           considerably cuts down on computation time.

SIZE=input('Enter an odd number of rows for grid.');
CELLS = input('Enter the number of iterations to be
           performed.');
A=zeros(SIZE,SIZE);
CENTER = SIZE/2+0.5;
A(CENTER,CENTER) = RED;
LEFT = CENTER-1;
RIGHT = CENTER+1;
for LOOP = 1:CELLS;
   CHOOSE4;
   pause(5)
end
RESET;
%This loop counts the number of RED cells
for i=LEFT:RIGHT
   for j=LEFT:RIGHT
      if A(i,j) > RED
         COUNT = COUNT + 1;
      end
   end
end
COUNT
% Error checking outputs
%LEFT
%RIGHT
%A
```

YELLOWDIST.m

```
% Convert YELLOW cells to a cumulative probability distribution
for i = LEFT:RIGHT
   for j = LEFT:RIGHT
      if A(i,j)<RED & A(i,j)>BLUE          %If it is YELLOW
         A(i,j)=(PROB*A(i,j))+LAST;        %Compute cumulative
                                             probability of cell
         LAST = A(i,j);                    %Track last
                                             probability to sum
      end
   end
end
```

MATLAB Code for Tumor5

ASSIGNLEFT.m See Tumor4.

ASSIGNRIGHT.m See Tumor4.

CHECKDIR.m

```
if A(i+ROW,j+COL)~=RED & A(i+ROW,j+COL)~= YELLOW
   A(i+ROW,j+COL) = YELLOW;
   ABS1 = abs((i+ROW)-CENTER);
   ABS2 = abs((j+COL)-CENTER);
        if ABS1 > ABS2
           ABSOLUTE = ABS1;
        else
           ABSOLUTE = ABS2;
        end
   SETTING = FILLED;
else
   ROW = ROW + (1 * sign(ROW));
   COL = COL + (1 * sign(COL));
   CHECKDIR;
end
```

CHOOSE5.m

```
SETTING = UNFILLED;
for i=LEFT:RIGHT
   for j=LEFT:RIGHT
      if A(i,j)==RED
         DIRECTION;
         while SETTING == UNFILLED
            CHECKDIR;
         end
         if ABSOLUTE > ABSMAX
            ABSMAX = ABSOLUTE;
            ASSIGNLEFT;
            ASSIGNRIGHT;
         end
      end
   SETTING = UNFILLED;
   end
end
image(uint8(A));
```

DIRECTION.m

```
RANDOM=rand;
if RANDOM < 0.125
   ROW = -1;
   COL = -1;
   %DIR =  1;
```

```
elseif RANDOM < 0.25
   ROW = -1;
   COL =  0;
   %DIR =  2;
elseif RANDOM < 0.375
   ROW = -1;
   COL =  1;
   %DIR =  3;
elseif RANDOM < 0.5
   ROW =  0;
   COL = -1;
   %DIR =  4;
elseif RANDOM < 0.625
   ROW =  0;
   COL =  1;
   %DIR =  5;
elseif RANDOM < 0.75
   ROW =  1;
   COL = -1;
   %DIR =  6;
elseif RANDOM < 0.875
   ROW =  1;
   COL =  0;
   %DIR =  7;
elseif RANDOM < 1
   ROW =  1;
   COL =  1;
   %DIR =  8;
end
%DIR
```

DIRECTION2.m

```
RANDOM=rand;
if RANDOM < 0.25
   ROW = 0;
   COL = 1;
   %DIR =  1;
elseif RANDOM < 0.5
   ROW =  0;
   COL = -1;
   %DIR =  2;
elseif RANDOM < 0.75
   ROW = 1;
   COL = 0;
   %DIR =  3;
elseif RANDOM < 1.0
   ROW =  -1;
   COL =   0;
   %DIR =  4;
end
%DIR
```

SETRED.m

```
for i=LEFT:RIGHT
   for j=LEFT:RIGHT
      if A(i,j)==YELLOW
         A(i,j) = RED;
      end
   end
end
```

TOTALREDS.m See Tumor4.

Tumor5.m

```
LEFT = 0;              RIGHT = 0;
SETTING = 0;           FILLED = 1;
UNFILLED = 0;      SIZE = 0;
LOOP = 0;              ROW = 0;
COL = 0;           RED = 50;
YELLOW = 25;           ABSOLUTE = 0;
ABS1 = 0;              ABS2 = 0;
ABSMAX = 0;        NUMREDS = 1;
SIZE = input('Enter an odd number of rows for grid.');
CELLS = input('Enter the number of iterations to be
               performed.');
A=zeros(SIZE,SIZE);
CENTER = (SIZE/2+0.5);
A(CENTER,CENTER) = RED;
LEFT = CENTER - 1;
RIGHT = CENTER + 1;
%CENTER
for LOOP = 1:CELLS;
   TOTALREDS;
   CHOOSE5;
   pause(0.5)
   SETRED;
end
image(uint8(A));
TOTALREDS;
NUMREDS
```

MATLAB Code for Tumor6

ASSIGNLEFT.m

```
if LEFT <= 1
   LEFT = 1;
else
   LEFT = LEFT - 1;
end
```

ASSIGNRIGHT.m

```
if RIGHT >= SIZE
   RIGHT = SIZE;
else
   RIGHT = RIGHT + 1;
end
```

BOUNDARY.m

```
%A(i+ROWCUM,j+COLCUM)
%i+ROWCUM
%j+COLCUM
%A(i+ROWCUM,j+COLCUM)=A(i+ROWCUM,j+COLCUM)+1;
if A(i+ROWCUM,j+COLCUM)>=HLOW & A(i+ROWCUM,j+COLCUM)<=HHIGH
   BOUNDARYTYPE = HORIZONTAL;
elseif A(i+ROWCUM,j+COLCUM)>=VLOW & A(i+ROWCUM,j+COLCUM)<=VHIGH
   BOUNDARYTYPE = VERTICAL;
elseif A(i+ROWCUM,j+COLCUM)>HHIGH & A(i+ROWCUM,j+COLCUM)<VLOW
   A(i+ROWCUM,j+COLCUM)=BLUE;
elseif A(i+ROWCUM,j+COLCUM)>VHIGH
   A(i+ROWCUM,j+COLCUM)=BLUE;
end
if BOUNDARYTYPE==HORIZONTAL
      if DELTAROW == ROW1 & DELTACOL == COL1
         DELTAROW = ROW6;
         DELTACOL = COL6;
         DIR = 6;
      elseif DELTAROW == ROW2 & DELTACOL == COL2
         RANDOM2 = rand;
         if RANDOM2 < 0.33333
            DELTAROW = ROW6;
            DELTACOL = COL6;
            DIR = 6;
         elseif RANDOM2 < 0.66666
            DELTAROW = ROW7;
            DELTACOL = COL7;
            DIR = 7;
         elseif RANDOM2 < 1.0
            DELTAROW = ROW8;
            DELTACOL = COL8;
            DIR = 8;
         end
      elseif DELTAROW == ROW3 & DELTACOL == COL3
         DELTAROW = ROW8;
         DELTACOL = COL8;
         DIR = 8;
      elseif DELTAROW == ROW6 & DELTACOL == COL6
         DELTAROW = ROW1;
         DELTACOL = COL1;
         DIR = 1;
      elseif DELTAROW == ROW7 & DELTACOL == COL7
         RANDOM3 = rand;
         if RANDOM3 < 0.33333
            DELTAROW = ROW1;
```

```
            DELTACOL = COL1;
            DIR = 1;
         elseif RANDOM3 < 0.66666
            DELTAROW = ROW2;
            DELTACOL = COL2;
            DIR = 2;
         elseif RANDOM3 < 1.0
            DELTAROW = ROW3;
            DELTACOL = COL3;
            DIR = 3;
         end
      elseif DELTAROW == ROW8 & DELTACOL == COL8
         DELTAROW = ROW3;
         DELTACOL = COL3;
         DIR = 3;
      end
elseif BOUNDARYTYPE==VERTICAL
      if DELTAROW == ROW1 & DELTACOL == COL1
         DELTAROW = ROW3;
         DELTACOL = COL3;
         DIR = 3;
      elseif DELTAROW == ROW4 & DELTACOL == COL4
         RANDOM4 = rand;
         if RANDOM4 < 0.33333
            DELTAROW = ROW3;
            DELTACOL = COL3;
            DIR = 3;
         elseif RANDOM4 < 0.66666
            DELTAROW = ROW5;
            DELTACOL = COL5;
            DIR = 5;
         elseif RANDOM4 < 1.0
            DELTAROW = ROW8;
            DELTACOL = COL8;
            DIR = 8;
         end
      elseif DELTAROW == ROW6 & DELTACOL == COL6
         DELTAROW = ROW8;
         DELTACOL = COL8;
         DIR = 8;
      elseif DELTAROW == ROW3 & DELTACOL == COL3
         DELTAROW = ROW1;
         DELTACOL = COL1;
         DIR = 1;
      elseif DELTAROW == ROW5 & DELTACOL == COL5
         RANDOM5 = rand;
         if RANDOM5 < 0.33333
            DELTAROW = ROW1;
            DELTACOL = COL1;
            DIR = 1;
         elseif RANDOM5 < 0.66666
            DELTAROW = ROW4;
            DELTACOL = COL4;
            DIR = 4;
         elseif RANDOM5 < 1.0
```

```
            DELTAROW = ROW6;
            DELTACOL = COL6;
            DIR = 6;
         end
      elseif DELTAROW == ROW8 & DELTACOL == COL8
         DELTAROW = ROW6;
         DELTACOL = COL6;
         DIR = 6;
      end
end
%DIR
CHECKDIR;
```

CHECKDIR.m

```
if A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL)>=HLOW
   ROWCUM = ROWCUM + DELTAROW;
   COLCUM = COLCUM + DELTACOL;
   BOUNDARY;
elseif A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL)~=RED &
                  A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL)~=YELLOW
   A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL) = YELLOW;
   ABS1 = abs((i+ROWCUM+DELTAROW)-CENTER);
   ABS2 = abs((j+COLCUM+DELTACOL)-CENTER);
        if ABS1 > ABS2
           ABSOLUTE = ABS1;
        else
           ABSOLUTE = ABS2;
        end
   SETTING = FILLED;
   %i+ROWCUM+DELTAROW
   %j+COLCUM+DELTACOL
else
   ROWCUM = ROWCUM + DELTAROW;
   COLCUM = COLCUM + DELTACOL;
   CHECKDIR;
end
```

CHOOSE6.m

```
SETTING = UNFILLED;
for i=LEFT:RIGHT
   for j=LEFT:RIGHT
      if A(i,j)==RED
         %i
         %j
         DIRECTION;
         while SETTING == UNFILLED
            CHECKDIR;
         end
         if ABSOLUTE > ABSMAX
            ABSMAX = ABSOLUTE;
            ASSIGNLEFT;
            ASSIGNRIGHT;
```

```
            %LEFT
            %RIGHT
         end
      end
      SETTING = UNFILLED;
      ROWCUM = 0;
      COLCUM = 0;
   end
end
image(uint8(A));
```

DIRECTION.m

```
    ROW1 = -1;    COL1 = -1;
   ROW2 = -1;    COL2 =  0;
   ROW3 = -1;    COL3 =  1;
   ROW4 =  0;    COL4 = -1;
   ROW5 =  0;    COL5 =  1;
   ROW6 =  1;    COL6 = -1;
   ROW7 =  1;    COL7 =  0;
   ROW8 =  1;    COL8 =  1;
RANDOM=rand;
if RANDOM < 0.125
   DELTAROW = ROW1;
   DELTACOL = COL1;
   DIR =  1;
elseif RANDOM < 0.25
   DELTAROW = ROW2;
   DELTACOL = COL2;
   DIR =  2;
elseif RANDOM < 0.375
   DELTAROW = ROW3;
   DELTACOL = COL3;
   DIR =  3;
elseif RANDOM < 0.5
   DELTAROW = ROW4;
   DELTACOL = COL4;
   DIR =  4;
elseif RANDOM < 0.625
   DELTAROW = ROW5;
   DELTACOL = COL5;
   DIR =  5;
elseif RANDOM < 0.75
   DELTAROW = ROW6;
   DELTACOL = COL6;
   DIR =  6;
elseif RANDOM < 0.875
   DELTAROW = ROW7;
   DELTACOL = COL7;
   DIR =  7;
elseif RANDOM < 1
   DELTAROW = ROW8;
   DELTACOL = COL8;
   DIR =  8;
end
```

```
%DIR
```

DIRECTION2.m

```
RANDOM=rand;
if RANDOM < 0.25
   ROW = 0;
   COL = 1;
   %DIR =  1;
elseif RANDOM < 0.5
   ROW =  0;
   COL = -1;
   %DIR =  2;
elseif RANDOM < 0.75
   ROW = 1;
   COL = 0;
   %DIR =  3;
elseif RANDOM < 1.0
   ROW =  -1;
   COL =   0;
   %DIR =  4;
end
%DIR
```

SETRED.m

```
for i=LEFT:RIGHT
   for j=LEFT:RIGHT
      if A(i,j)==YELLOW
         A(i,j) = RED;
      end
   end
end
```

TOTALREDS.m

```
NUMREDS = 0;
for i = LEFT:RIGHT
   for j = LEFT:RIGHT
      if A(i,j) == RED
         NUMREDS = NUMREDS + 1;
      end
   end
end
%NUMREDS
```

Tumor6.m

```
function B = Tumor6(SIZE,CELLS)

LEFT = 0;              RIGHT = 0;        SETTING = 0;
FILLED = 1;          UNFILLED = 0;     LOOP = 0;
RED = 50;              YELLOW = 25;        ABSOLUTE = 0;
ABS1 = 0;              ABS2 = 0;           ABSMAX = 0;
NUMREDS = 1;           BLUE = 0;           BOUNDARYTYPE = 0;
```

```
VERTICAL = 1;          HORIZONTAL = 0;    DELTAROW = 0;
DELTACOL = 0;          ROWCUM = 0;        COLCUM = 0;
HLOW   = 140;          HHIGH  = 144;      VLOW   = 200;
VHIGH  = 230;

%SIZE=input('Enter an odd number of rows for grid.');
%CELLS = input('Enter the number of iterations to be
                performed.');
A=zeros(SIZE,SIZE);
CENTER = (SIZE/2+0.5);
A(CENTER,CENTER) = RED;
LEFT = CENTER - 1;
RIGHT = CENTER + 1;
%CENTER
%for d=1:SIZE
%   A(d,CENTER-5)=VLOW;
%   A(d,CENTER+5)=VLOW;
%end
for e=1:SIZE
   A(CENTER-5,e)=HLOW;
   A(CENTER+5,e)=HLOW;
end
for LOOP = 1:CELLS;
   TOTALREDS;
   CHOOSE6;
   pause(0.5)
   SETRED;
end
image(uint8(A));
TOTALREDS;
NUMREDS
B=size(A);
```

MATLAB Code for Tumor7

ASSIGNLEFT.m See Tumor6.

ASSIGNRIGHT.m See Tumor6.

BOUNDARY.m

```
%A(i+ROWCUM,j+COLCUM)
%i+ROWCUM
%j+COLCUM
A(i+ROWCUM,j+COLCUM)=A(i+ROWCUM,j+COLCUM)+1;

if A(i+ROWCUM,j+COLCUM)>=HLOW & A(i+ROWCUM,j+COLCUM)<=HHIGH
   BOUNDARYTYPE = HORIZONTAL;
elseif A(i+ROWCUM,j+COLCUM)>=VLOW & A(i+ROWCUM,j+COLCUM)<=VHIGH
   BOUNDARYTYPE = VERTICAL;
elseif A(i+ROWCUM,j+COLCUM)>HHIGH & A(i+ROWCUM,j+COLCUM)<VLOW
   A(i+ROWCUM,j+COLCUM)=BLUE;
elseif A(i+ROWCUM,j+COLCUM)>VHIGH
```

```
   A(i+ROWCUM,j+COLCUM)=BLUE;
end

if BOUNDARYTYPE==HORIZONTAL
      if DELTAROW == ROW1 & DELTACOL == COL1
         DELTAROW = ROW6;
         DELTACOL = COL6;
         DIR = 6;
      elseif DELTAROW == ROW2 & DELTACOL == COL2
         RANDOM2 = rand;
         if RANDOM2 < 0.33333
            DELTAROW = ROW6;
            DELTACOL = COL6;
            DIR = 6;
         elseif RANDOM2 < 0.66666
            DELTAROW = ROW7;
            DELTACOL = COL7;
            DIR = 7;
         elseif RANDOM2 < 1.0
            DELTAROW = ROW8;
            DELTACOL = COL8;
            DIR = 8;
         end
      elseif DELTAROW == ROW3 & DELTACOL == COL3
         DELTAROW = ROW8;
         DELTACOL = COL8;
         DIR = 8;
      elseif DELTAROW == ROW6 & DELTACOL == COL6
         DELTAROW = ROW1;
         DELTACOL = COL1;
         DIR = 1;
      elseif DELTAROW == ROW7 & DELTACOL == COL7
         RANDOM3 = rand;
         if RANDOM3 < 0.33333
            DELTAROW = ROW1;
            DELTACOL = COL1;
            DIR = 1;
         elseif RANDOM3 < 0.66666
            DELTAROW = ROW2;
            DELTACOL = COL2;
            DIR = 2;
         elseif RANDOM3 < 1.0
            DELTAROW = ROW3;
            DELTACOL = COL3;
            DIR = 3;
         end
      elseif DELTAROW == ROW8 & DELTACOL == COL8
         DELTAROW = ROW3;
         DELTACOL = COL3;
         DIR = 3;
      end
elseif BOUNDARYTYPE==VERTICAL
      if DELTAROW == ROW1 & DELTACOL == COL1
         DELTAROW = ROW3;
         DELTACOL = COL3;
```

```
      DIR = 3;
   elseif DELTAROW == ROW4 & DELTACOL == COL4
      RANDOM4 = rand;
      if RANDOM4 < 0.33333
         DELTAROW = ROW3;
         DELTACOL = COL3;
         DIR = 3;
      elseif RANDOM4 < 0.66666
         DELTAROW = ROW5;
         DELTACOL = COL5;
         DIR = 5;
      elseif RANDOM4 < 1.0
         DELTAROW = ROW8;
         DELTACOL = COL8;
         DIR = 8;
      end
   elseif DELTAROW == ROW6 & DELTACOL == COL6
      DELTAROW = ROW8;
      DELTACOL = COL8;
      DIR = 8;
   elseif DELTAROW == ROW3 & DELTACOL == COL3
      DELTAROW = ROW1;
      DELTACOL = COL1;
      DIR = 1;
   elseif DELTAROW == ROW5 & DELTACOL == COL5
      RANDOM5 = rand;
      if RANDOM5 < 0.33333
         DELTAROW = ROW1;
         DELTACOL = COL1;
         DIR = 1;
      elseif RANDOM5 < 0.66666
         DELTAROW = ROW4;
         DELTACOL = COL4;
         DIR = 4;
      elseif RANDOM5 < 1.0
         DELTAROW = ROW6;
         DELTACOL = COL6;
         DIR = 6;
      end
   elseif DELTAROW == ROW8 & DELTACOL == COL8
      DELTAROW = ROW6;
      DELTACOL = COL6;
      DIR = 6;
   end
end
%DIR
CHECKDIR;
```

CHECKDIR.m See Tumor6

CHOOSE7.m See Tumor6

DIRECTION.m See Tumor6.

DIRECTION2.m See Tumor6.

SETRED.m See Tumor6.

TOTALREDS.m See Tumor6.

Tumor7.m

```
function B = Tumor7(SIZE,CELLS)

LEFT = 0;              RIGHT = 0;          SETTING = 0;
FILLED = 1;                  UNFILLED = 0;      LOOP = 0;
RED = 50;                    YELLOW = 25;          ABSOLUTE = 0;
ABS1 = 0;                    ABS2 = 0;             ABSMAX = 0;
NUMREDS = 1;                 BLUE = 0;             BOUNDARYTYPE = 0;
VERTICAL = 1;              HORIZONTAL = 0;    DELTAROW = 0;
DELTACOL = 0;              ROWCUM = 0;        COLCUM = 0;
HLOW   = 140;              HHIGH  = 144;      VLOW   = 200;
VHIGH  = 230;

%SIZE=input('Enter an odd number of rows for grid.');
%CELLS = input('Enter the number of iterations to be
                    performed.');
A=zeros(SIZE,SIZE);
CENTER = (SIZE/2+0.5);
A(CENTER,CENTER) = RED;
LEFT = CENTER - 1;
RIGHT = CENTER + 1;
%CENTER
%for d=1:SIZE
%   A(d,CENTER-5)=VLOW;
%   A(d,CENTER+5)=VLOW;
%end
for e=1:SIZE
   A(CENTER-5,e)=HLOW;
   A(CENTER+5,e)=HLOW;
end
for LOOP = 1:CELLS;
   TOTALREDS;
   CHOOSE7;
   pause(0.5)
   SETRED;
end
image(uint8(A));
TOTALREDS;
NUMREDS
B=size(A);
```

Appendix B – MATLAB Code for 3-Dimensional Models

There are three 3-dimensional models created for use by this thesis effort. The following table lists which subroutines are executed with each model, what each model accomplishes, and what command runs the model. It is important to note that when running these models, MATLAB will return an error if the MATLAB directory is not pointed at the directory that contains all of the subroutines that are associated with the model to be executed. Also, it is imperative to understand the limitations of each MATLAB package. For example, the Student Version of MATLAB (Version 5.0.0.4073) only allows a 25 x 25 x 25 grid to be created in three dimensions. Also, any population doubles (iterations) greater than 12 may overwhelm the computer memory. The 3-dimensional models that create boundaries are less volatile than their 2-dimensional cousins, but are more apt to failure than the 3-dimensional models that do not create boundaries. Again, this is due to the fact that the algorithm used for tumor-boundary interactions is improved in the 3-dimensional version.

Following Table B-1 is the MATLAB code for each model. Comments begin with a "%". Any commented MATLAB commands, i.e. not sentences, are merely commands that were used to check for errors or changes in code.

Table B-1. 3-Dimensional Model Information.

Model	Associated Subroutines	Execution Command	Model Output
Tumor3d	ASSIGNBACK.m ASSIGNBACK1.m ASSIGNBOTTOM.m ASSIGNBOTTOM1.m ASSIGNFRONT.m ASSIGNFRONT1.m ASSIGNLEFT.m ASSIGNLEFT1.m ASSIGNRIGHT.m ASSIGNRIGHT1.m ASSIGNTOP.m ASSIGNTOP1.m CHECKDIR.m CHOOSE3d.m DIRECTION.m SETRED.m TOTALREDS.m Tumor3d.m VIEWTUMOR.m	>>*Tumor3d(Size, Cells, X-Position, Y-Position, Z-Position)*	A *Size* x *Size* x *Size* image that contains 2^{Cells} cells within it. The first malignant cell is situated at the point (*X-Position, Y-Position, Z-Position*). Tumor3d uses the same algorithm as the latter 2-dimensional models.
Tumor3d2	ASSIGNBACK.m ASSIGNBACK1.m ASSIGNBOTTOM.m ASSIGNBOTTOM1.m ASSIGNFRONT.m ASSIGNFRONT1.m ASSIGNLEFT.m ASSIGNLEFT1.m	>>*Tumor3d2(Size, Cells, X-Position, Y-Position, Z-Position)*	A *Size* x *Size* x *Size* image that contains 2^{Cells} cells within it, where the first cell is situated at (*X-Position, Y-Position, Z-Position*). Two

	ASSIGNRIGHT.m ASSIGNRIGHT1.m ASSIGNTOP.m ASSIGNTOP1.m CHECKDIR.m CHOOSE3d2.m DIRECTION.m SETRED.m TOTALREDS.m Tumor3d2.m VIEWTUMOR.m		impenetrable boundary planes are created. They are both looked "through" when viewing from the Overhead viewpoint.
Tumor3d3	ASSIGNBACK.m ASSIGNBACK1.m ASSIGNBOTTOM.m ASSIGNBOTTOM1.m ASSIGNFRONT.m ASSIGNFRONT1.m ASSIGNLEFT.m ASSIGNLEFT1.m ASSIGNRIGHT.m ASSIGNRIGHT1.m ASSIGNTOP.m ASSIGNTOP1.m CHECKDIR.m CHOOSE3d3.m DIRECTION.m SETRED.m TOTALREDS.m Tumor3d3.m VIEWTUMOR.m	*>>Tumor3d3(Size, Cells, X-Position, Y-Position, Z-Position)*	A *Size* x *Size* x *Size* image that contains 2^{Cells} cells within it, where the first cell is situated at (*X-Position, Y-Position, Z-Position*). Also created is a penetrable boundary that represents a duct-like breast structure. The Front viewpoint shows the cross-section of the duct.

MATLAB Code for Tumor3d

ASSIGNBACK.m

```
if BACK >= SIZE
   BACK = SIZE;
else
   BACK = BACK;
end
```

ASSIGNBACK1.m

```
if BACK1 >= SIZE
   BACK1 = SIZE;
else
   BACK1 = BACK1 + 1;
end
```

ASSIGNBOTTOM.m

```
if BOTTOM >= SIZE
   BOTTOM = SIZE;
else
   BOTTOM = BOTTOM;
end
```

ASSIGNBOTTOM1.m

```
if BOTTOM1 >= SIZE
   BOTTOM1 = SIZE;
else
   BOTTOM1 = BOTTOM1 + 1;
end
```

ASSIGNFRONT.m

```
if FRONT <= 1
   FRONT = 1;
else
   FRONT = FRONT;
end
```

ASSIGNFRONT1.m

```
if FRONT1 <= 1
   FRONT1 = 1;
else
   FRONT1 = FRONT1 - 1;
end
```

ASSIGNLEFT.m

```
if LEFT <= 1
   LEFT = 1;
```

```
else
   LEFT = LEFT;
end
```

ASSIGNLEFT1.m

```
if LEFT1 <= 1
   LEFT1 = 1;
else
   LEFT1 = LEFT1 - 1;
end
```

ASSIGNRIGHT.m

```
if RIGHT >= SIZE
   RIGHT = SIZE;
else
   RIGHT = RIGHT;
end
```

ASSIGNRIGHT1.m

```
if RIGHT1 >= SIZE
   RIGHT1 = SIZE;
else
   RIGHT1 = RIGHT1 + 1;
end
```

ASSIGNTOP.m

```
if TOP <= 1
   TOP = 1;
else
   TOP = TOP;
end
```

ASSIGNTOP1.m

```
if TOP1 <= 1
   TOP1 = 1;
else
   TOP1 = TOP1 - 1;
end
```

CHECKDIR.m

```
if i+ROWCUM+DELTAROW < 1 | j+COLCUM+DELTACOL < 1 | k+DEPCUM+DELTADEP <1
   DIRECTION;
   CHECKDIR;
elseif i+ROWCUM+DELTAROW > SIZE | j+COLCUM+DELTACOL > SIZE |
k+DEPCUM+DELTADEP > SIZE
   DIRECTION;
   CHECKDIR;
```

```
elseif A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL,k+DEPCUM+DELTADEP)~=RED &
A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL,k+DEPCUM+DELTADEP)~=YELLOW
   A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL,k+DEPCUM+DELTADEP) = YELLOW;
   ABS1 = abs((i+ROWCUM+DELTAROW)-X);
   ABS2 = abs((j+COLCUM+DELTACOL)-Y);
   ABS3 = abs((k+DEPCUM+DELTADEP)-Z);
   VECTOR = [ABS1,ABS2,ABS3];
   ABSOLUTE = max(VECTOR);
   SETTING = FILLED;
else
   ROWCUM = ROWCUM + DELTAROW;
   COLCUM = COLCUM + DELTACOL;
   DEPCUM = DEPCUM + DELTADEP;
   CHECKDIR;
end
```

CHOOSE3d.m

```
SETTING = UNFILLED;
for i=LEFT:RIGHT
   for j=TOP:BOTTOM
      for k = FRONT:BACK
         if A(i,j,k)==RED
            DIRECTION;
            while SETTING == UNFILLED
               CHECKDIR;
            end
            if ABSOLUTE > ABSMAX
               ABSMAX = ABSOLUTE;
               ASSIGNLEFT1;
               ASSIGNRIGHT1;
               ASSIGNTOP1;
               ASSIGNBOTTOM1;
               ASSIGNFRONT1;
               ASSIGNBACK1;
            end
         end
         SETTING = UNFILLED;
         ROWCUM  = 0;
         COLCUM  = 0;
         DEPCUM  = 0;
      end
   end
end
LEFT   = LEFT1;
RIGHT  = RIGHT1;
TOP    = TOP1;
BOTTOM = BOTTOM1;
BACK   = BACK1;
FRONT  = FRONT1;
```

DIRECTION.m

```
ROW1 = -1;  COL1 =  -1;  DEP1  = -1;
ROW2 = -1;  COL2 =  -1;  DEP2  =  0;
```

```
ROW3 = -1;  COL3 =  -1;  DEP3  =  1;
ROW4 = -1;  COL4 =   0;  DEP4  = -1;
ROW5 = -1;  COL5 =   0;  DEP5  =  0;
ROW6 = -1;  COL6 =   0;  DEP6  =  1;
ROW7 = -1;  COL7 =   1;  DEP7  = -1;
ROW8 = -1;  COL8 =   1;  DEP8  =  0;
ROW9 = -1;  COL9 =   1;  DEP9  =  1;
ROW10 = 0;  COL10 = -1;  DEP10 = -1;
ROW11 = 0;  COL11 = -1;  DEP11 =  0;
ROW12 = 0;  COL12 = -1;  DEP12 =  1;
ROW13 = 0;  COL13 =  0;  DEP13 = -1;
ROW14 = 0;  COL14 =  0;  DEP14 =  1;
ROW15 = 0;  COL15 =  1;  DEP15 = -1;
ROW16 = 0;  COL16 =  1;  DEP16 =  0;
ROW17 = 0;  COL17 =  1;  DEP17 =  1;
ROW18 = 1;  COL18 = -1;  DEP18 = -1;
ROW19 = 1;  COL19 = -1;  DEP19 =  0;
ROW20 = 1;  COL20 = -1;  DEP20 =  1;
ROW21 = 1;  COL21 =  0;  DEP21 = -1;
ROW22 = 1;  COL22 =  0;  DEP22 =  0;
ROW23 = 1;  COL23 =  0;  DEP23 =  1;
ROW24 = 1;  COL24 =  1;  DEP24 = -1;
ROW25 = 1;  COL25 =  1;  DEP25 =  0;
ROW26 = 1;  COL26 =  1;  DEP26 =  1;

RANDOM=rand;

if RANDOM < 1/26
   DELTAROW = ROW1;
   DELTACOL = COL1;
   DELTADEP = DEP1;
   DIR =  1;
elseif RANDOM < 2/26
   DELTAROW = ROW2;
   DELTACOL = COL2;
   DELTADEP = DEP2;
   DIR =  2;
elseif RANDOM < 3/26
   DELTAROW = ROW3;
   DELTACOL = COL3;
   DELTADEP = DEP3;
   DIR =  3;
elseif RANDOM < 4/26
   DELTAROW = ROW4;
   DELTACOL = COL4;
   DELTADEP = DEP4;
   DIR =  4;
elseif RANDOM < 5/26
   DELTAROW = ROW5;
   DELTACOL = COL5;
   DELTADEP = DEP5;
   DIR =  5;
elseif RANDOM < 6/26
   DELTAROW = ROW6;
   DELTACOL = COL6;
```

```
   DELTADEP = DEP6;
   DIR =  6;
elseif RANDOM < 7/26
   DELTAROW = ROW7;
   DELTACOL = COL7;
   DELTADEP = DEP7;
   DIR =  7;
elseif RANDOM < 8/26
   DELTAROW = ROW8;
   DELTACOL = COL8;
   DELTADEP = DEP8;
   DIR =  8;
elseif RANDOM < 9/26
   DELTAROW = ROW9;
   DELTACOL = COL9;
   DELTADEP = DEP9;
   DIR =  9;
elseif RANDOM < 10/26
   DELTAROW = ROW10;
   DELTACOL = COL10;
   DELTADEP = DEP10;
   DIR =  10;
elseif RANDOM < 11/26
   DELTAROW = ROW11;
   DELTACOL = COL11;
   DELTADEP = DEP11;
   DIR =  11;
elseif RANDOM < 12/26
   DELTAROW = ROW12;
   DELTACOL = COL12;
   DELTADEP = DEP12;
   DIR =  12;
elseif RANDOM < 13/26
   DELTAROW = ROW13;
   DELTACOL = COL13;
   DELTADEP = DEP13;
   DIR =  13;
elseif RANDOM < 14/26
   DELTAROW = ROW14;
   DELTACOL = COL14;
   DELTADEP = DEP14;
   DIR =  14;
elseif RANDOM < 15/26
   DELTAROW = ROW15;
   DELTACOL = COL15;
   DELTADEP = DEP15;
   DIR =  15;
elseif RANDOM < 16/26
   DELTAROW = ROW16;
   DELTACOL = COL16;
   DELTADEP = DEP16;
   DIR =  16;
elseif RANDOM < 17/26
   DELTAROW = ROW17;
   DELTACOL = COL17;
```

```
   DELTADEP = DEP17;
   DIR =  17;
elseif RANDOM < 18/26
   DELTAROW = ROW18;
   DELTACOL = COL18;
   DELTADEP = DEP18;
   DIR =  18;
elseif RANDOM < 19/26
   DELTAROW = ROW19;
   DELTACOL = COL19;
   DELTADEP = DEP19;
   DIR =  19;
elseif RANDOM < 20/26
   DELTAROW = ROW20;
   DELTACOL = COL20;
   DELTADEP = DEP20;
   DIR =  20;
elseif RANDOM < 21/26
   DELTAROW = ROW21;
   DELTACOL = COL21;
   DELTADEP = DEP21;
   DIR =  21;
elseif RANDOM < 22/26
   DELTAROW = ROW22;
   DELTACOL = COL22;
   DELTADEP = DEP22;
   DIR =  22;
elseif RANDOM < 23/26
   DELTAROW = ROW23;
   DELTACOL = COL23;
   DELTADEP = DEP23;
   DIR =  23;
elseif RANDOM < 24/26
   DELTAROW = ROW24;
   DELTACOL = COL24;
   DELTADEP = DEP24;
   DIR =  24;
elseif RANDOM < 25/26
   DELTAROW = ROW25;
   DELTACOL = COL25;
   DELTADEP = DEP25;
   DIR =  25;
elseif RANDOM < 26/26
   DELTAROW = ROW26;
   DELTACOL = COL26;
   DELTADEP = DEP26;
   DIR =  26;
end

%DIR
%DELTAROW
%DELTACOL
%DELTADEP
```

SETRED.m

```
for i=LEFT:RIGHT
   for j=TOP:BOTTOM
      for k = FRONT:BACK
         if A(i,j,k)==YELLOW
            A(i,j,k) = RED;
         end
      end
   end
end
```

TOTALREDS.m

```
NUMREDS = 0;
for i = LEFT:RIGHT
   for j = TOP:BOTTOM
      for k = FRONT:BACK
         if A(i,j,k) == RED
            NUMREDS = NUMREDS + 1;
         end
      end
   end
end
NUMREDS
```

Tumor3d.m

```
function B = Tumor3d(SIZE,CELLS,X,Y,Z)

LEFT     = 0;              RIGHT    = 0;
TOP      = 0;              BOTTOM   = 0;
FRONT    = 0;              BACK     = 0;
FILLED   = 1;              UNFILLED = 0;
SETTING  = UNFILLED;       LOOP     = 0;
RED      = 4;              YELLOW   = 1;
BLUE     = 0;              NUMREDS  = 1;
DELTAROW = 0;              DELTACOL = 0;
DELTADEP = 0;              ROWCUM   = 0;
COLCUM   = 0;              DEPCUM   = 0;
ABSOLUTE = 0;              ABSMAX   = 0;

A = zeros(SIZE,SIZE,SIZE);
A(X,Y,Z) = RED;
LEFT    = X - 1;       RIGHT   = X + 1;
TOP     = Y - 1;       BOTTOM  = Y + 1;
FRONT   = Z - 1;       BACK    = Z + 1;
ASSIGNLEFT;    ASSIGNRIGHT;
ASSIGNTOP;     ASSIGNBOTTOM;
ASSIGNFRONT;      ASSIGNBACK;
LEFT1   = LEFT;       RIGHT1  = RIGHT;
TOP1    = TOP;        BOTTOM1 = BOTTOM;
BACK1   = BACK;       FRONT1  = FRONT;
for LOOP = 1:CELLS;
   CHOOSE3d;
```

```
   SETRED;
end
VIEWTUMOR;
TOTALREDS;
B=size(A);
```

VIEWTUMOR.m

```
clear figure
C=sum(A,1);                %Side View
F=sum(A,2);                %Overhead View
H=sum(A,3);                %Front View

l = 0;
m = 0;
D=zeros(SIZE);
for l = 1:SIZE
   D = cat(1,D,C(:,:,l));
end
E = D(SIZE+1:2*SIZE,1:SIZE);

G=F(:,:,1);
for m = 2:SIZE
   G = cat(2,G,F(:,:,m));
end

figure(1)
hold on;
image(uint8(E));
title('Side View')
hold off;

figure(2)
hold on;
image(uint8(G));
title('Overhead View')
hold off;

figure(3)
hold on;
image(uint8(H));
title('Front View')
hold off;
```

MATLAB Code for Tumor3d2

ASSIGNBACK.m See Tumor3d.

ASSIGNBACK1.m See Tumor3d.

ASSIGNBOTTOM.m See Tumor3d.

ASSIGNBOTTOM1.m See Tumor3d.

ASSIGNFRONT.m See Tumor3d.

ASSIGNFRONT1.m See Tumor3d.

ASSIGNLEFT.m See Tumor3d.

ASSIGNLEFT1.m See Tumor3d.

ASSIGNRIGHT.m See Tumor3d.

ASSIGNRIGHT1.m See Tumor3d.

ASSIGNTOP.m See Tumor3d.

ASSIGNTOP1.m See Tumor3d.

CHECKDIR.m

```
if i+ROWCUM+DELTAROW < 1 | j+COLCUM+DELTACOL < 1 | k+DEPCUM+DELTADEP <1
   DIRECTION;
   CHECKDIR;
elseif i+ROWCUM+DELTAROW > SIZE | j+COLCUM+DELTACOL > SIZE |
k+DEPCUM+DELTADEP > SIZE
   DIRECTION;
   CHECKDIR;
elseif A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL,k+DEPCUM+DELTADEP) == BLUE
   A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL,k+DEPCUM+DELTADEP) = YELLOW;
   ABS1 = abs((i+ROWCUM+DELTAROW)-X);
   ABS2 = abs((j+COLCUM+DELTACOL)-Y);
   ABS3 = abs((k+DEPCUM+DELTADEP)-Z);
   VECTOR = [ABS1,ABS2,ABS3];
   ABSOLUTE = max(VECTOR);
   SETTING = FILLED;
elseif A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL,k+DEPCUM+DELTADEP) ==
BLACK
   DIRECTION;
   CHECKDIR;
else
   ROWCUM = ROWCUM + DELTAROW;
   COLCUM = COLCUM + DELTACOL;
   DEPCUM = DEPCUM + DELTADEP;
   CHECKDIR;
end
```

CHOOSE3d2.m See Tumor3d.

DIRECTION.m

```
%Sets up an assignment array for the value of the
%respective delta values for each direction.  The
%delta values are the incrementing values that
%tell the dividing cell which way to go.
ROW(1) = -1;  COL(1) =  -1;  DEP(1)  = -1;
ROW(2) = -1;  COL(2) =  -1;  DEP(2)  =  0;
ROW(3) = -1;  COL(3) =  -1;  DEP(3)  =  1;
ROW(4) = -1;  COL(4) =   0;  DEP(4)  = -1;
ROW(5) = -1;  COL(5) =   0;  DEP(5)  =  0;
```

```
ROW(6) = -1;  COL(6) =   0;  DEP(6)  =  1;
ROW(7) = -1;  COL(7) =   1;  DEP(7)  = -1;
ROW(8) = -1;  COL(8) =   1;  DEP(8)  =  0;
ROW(9) = -1;  COL(9) =   1;  DEP(9)  =  1;
ROW(10) = 0;  COL(10) = -1;  DEP(10) = -1;
ROW(11) = 0;  COL(11) = -1;  DEP(11) =  0;
ROW(12) = 0;  COL(12) = -1;  DEP(12) =  1;
ROW(13) = 0;  COL(13) =  0;  DEP(13) = -1;
ROW(14) = 0;  COL(14) =  0;  DEP(14) =  1;
ROW(15) = 0;  COL(15) =  1;  DEP(15) = -1;
ROW(16) = 0;  COL(16) =  1;  DEP(16) =  0;
ROW(17) = 0;  COL(17) =  1;  DEP(17) =  1;
ROW(18) = 1;  COL(18) = -1;  DEP(18) = -1;
ROW(19) = 1;  COL(19) = -1;  DEP(19) =  0;
ROW(20) = 1;  COL(20) = -1;  DEP(20) =  1;
ROW(21) = 1;  COL(21) =  0;  DEP(21) = -1;
ROW(22) = 1;  COL(22) =  0;  DEP(22) =  0;
ROW(23) = 1;  COL(23) =  0;  DEP(23) =  1;
ROW(24) = 1;  COL(24) =  1;  DEP(24) = -1;
ROW(25) = 1;  COL(25) =  1;  DEP(25) =  0;
ROW(26) = 1;  COL(26) =  1;  DEP(26) =  1;

%Generates a random number to select a direction and
%an opposite direction (which was tested but never implemented).
RANDOM=rand;
RANDOM2 = rand;

%Below is the algorithm for choosing a direction.  Notice that the
%Variables beginning with "OPP" are never invoked.  They were a test
case.
%So the random number selects a direction, which has the same
probability as
%every other direction.  Then assigns a delta value for the array.
if RANDOM < 1/26
   V = 1;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 26;
   elseif RANDOM2 < 0.4
      W = 23;
   elseif RANDOM2 < 0.6
      W = 25;
   elseif RANDOM2 < 0.8
      W = 22;
   else
      W = 17;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 2/26
   V = 2;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 24;
   elseif RANDOM2 < 0.4
      W = 25;
```

```
   elseif RANDOM2 < 0.6
      W = 26;
   elseif RANDOM2 < 0.8
      W = 22;
   else
      W = 16;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 3/26
   V = 3;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 21;
   elseif RANDOM2 < 0.4
      W = 22;
   elseif RANDOM2 < 0.6
      W = 24;
   elseif RANDOM2 < 0.8
      W = 25;
   else
      W = 15;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 4/26
   V = 4;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 14;
   elseif RANDOM2 < 0.4
      W = 20;
   elseif RANDOM2 < 0.6
      W = 22;
   elseif RANDOM2 < 0.8
      W = 23;
   else
      W = 26;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 5/26
   V = 5;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 19;
   elseif RANDOM2 < 0.4
      W = 21;
   elseif RANDOM2 < 0.6
      W = 22;
   elseif RANDOM2 < 0.8
      W = 23;
   else
      W = 25;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 6/26
```

```
   V = 6;  DELTAROW = ROW(V);    DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 13;
   elseif RANDOM2 < 0.4
      W = 18;
   elseif RANDOM2 < 0.6
      W = 21;
   elseif RANDOM2 < 0.8
      W = 22;
   else
      W = 24;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 7/26
   V = 7;  DELTAROW = ROW(V);    DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 12;
   elseif RANDOM2 < 0.4
      W = 19;
   elseif RANDOM2 < 0.6
      W = 20;
   elseif RANDOM2 < 0.8
      W = 22;
   else
      W = 23;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 8/26
   V = 8;  DELTAROW = ROW(V);    DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 11;
   elseif RANDOM2 < 0.4
      W = 18;
   elseif RANDOM2 < 0.6
      W = 19;
   elseif RANDOM2 < 0.8
      W = 20;
   else
      W = 22;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 9/26
   V = 9;  DELTAROW = ROW(V);    DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 10;
   elseif RANDOM2 < 0.4
      W = 18;
   elseif RANDOM2 < 0.6
      W = 19;
   elseif RANDOM2 < 0.8
      W = 21;
```

```
   else
      W = 22;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 10/26
   V = 10;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 9;
   elseif RANDOM2 < 0.4
      W = 17;
   elseif RANDOM2 < 0.6
      W = 26;
   elseif RANDOM2 < 0.8
      W = 14;
   else
      W = 16;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 11/26
   V = 11;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 8;
   elseif RANDOM2 < 0.4
      W = 15;
   elseif RANDOM2 < 0.6
      W = 16;
   elseif RANDOM2 < 0.8
      W = 17;
   else
      W = 25;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 12/26
   V = 12;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 7;
   elseif RANDOM2 < 0.4
      W = 13;
   elseif RANDOM2 < 0.6
      W = 15;
   elseif RANDOM2 < 0.8
      W = 16;
   else
      W = 24;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 13/26
   V = 13;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 6;
   elseif RANDOM2 < 0.4
```

```
      W = 12;
   elseif RANDOM2 < 0.6
      W = 14;
   elseif RANDOM2 < 0.8
      W = 17;
   else
      W = 23;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 14/26
   V = 14;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 4;
   elseif RANDOM2 < 0.4
      W = 10;
   elseif RANDOM2 < 0.6
      W = 13;
   elseif RANDOM2 < 0.8
      W = 15;
   else
      W = 21;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 15/26
   V = 15;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 3;
   elseif RANDOM2 < 0.4
      W = 11;
   elseif RANDOM2 < 0.6
      W = 12;
   elseif RANDOM2 < 0.8
      W = 14;
   else
      W = 20;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 16/26
   V = 16;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 2;
   elseif RANDOM2 < 0.4
      W = 10;
   elseif RANDOM2 < 0.6
      W = 11;
   elseif RANDOM2 < 0.8
      W = 12;
   else
      W = 19;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 17/26
```

```
   V = 17;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 1;
   elseif RANDOM2 < 0.4
      W = 10;
   elseif RANDOM2 < 0.6
      W = 18;
   elseif RANDOM2 < 0.8
      W = 11;
   else
      W = 13;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 18/26
   V = 18;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 5;
   elseif RANDOM2 < 0.4
      W = 6;
   elseif RANDOM2 < 0.6
      W = 8;
   elseif RANDOM2 < 0.8
      W = 9;
   else
      W = 17;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 19/26
   V = 19;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 5;
   elseif RANDOM2 < 0.4
      W = 7;
   elseif RANDOM2 < 0.6
      W = 8;
   elseif RANDOM2 < 0.8
      W = 9;
   else
      W = 16;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 20/26
   V = 20;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 4;
   elseif RANDOM2 < 0.4
      W = 5;
   elseif RANDOM2 < 0.6
      W = 7;
   elseif RANDOM2 < 0.8
      W = 8;
```

```
   else
      W = 15;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 21/26
   V = 21;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 3;
   elseif RANDOM2 < 0.4
      W = 5;
   elseif RANDOM2 < 0.6
      W = 6;
   elseif RANDOM2 < 0.8
      W = 9;
   else
      W = 14;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 22/26
   V = 22;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 2;
   elseif RANDOM2 < 0.4
      W = 4;
   elseif RANDOM2 < 0.6
      W = 5;
   elseif RANDOM2 < 0.8
      W = 6;
   else
      W = 8;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 23/26
   V = 23;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 1;
   elseif RANDOM2 < 0.4
      W = 4;
   elseif RANDOM2 < 0.6
      W = 7;
   elseif RANDOM2 < 0.8
      W = 13;
   else
      W = 5;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 24/26
   V = 24;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 2;
   elseif RANDOM2 < 0.4
```

```
      W = 3;
   elseif RANDOM2 < 0.6
      W = 5;
   elseif RANDOM2 < 0.8
      W = 6;
   else
      W = 12;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 25/26
   V = 25;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 1;
   elseif RANDOM2 < 0.4
      W = 2;
   elseif RANDOM2 < 0.6
      W = 3;
   elseif RANDOM2 < 0.8
      W = 5;
   else
      W = 11;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
elseif RANDOM < 26/26
   V = 26;  DELTAROW = ROW(V);   DELTACOL = COL(V);  DELTADEP = DEP(V);
DIR =  V;
   if RANDOM2 < 0.2
      W = 1;
   elseif RANDOM2 < 0.4
      W = 2;
   elseif RANDOM2 < 0.6
      W = 4;
   elseif RANDOM2 < 0.8
      W = 5;
   else
      W = 10;
   end
   OPPROW = ROW(W); OPPCOL = COL(W); OPPDEP = DEP(W); OPPDIR = W;
end
```

SETRED.m See Tumor3d.

TOTALREDS.m See Tumor3d.

Tumor3d2.m

```
function B = Tumor3d2(SIZE,CELLS,X,Y,Z)

DELTAROW = 0;   DELTACOL = 0;   DELTADEP = 0;
ROWCUM   = 0;   COLCUM   = 0;   DEPCUM   = 0;
LEFT     = 0;   RIGHT    = 0;
TOP      = 0;   BOTTOM   = 0;
FRONT    = 0;   BACK     = 0;
```

```
FILLED   = 1;   UNFILLED = 0;
BLACK    = 4;   RED      = 2;
YELLOW   = 1;   BLUE     = 0;
ABSOLUTE = 0;   ABSMAX   = 0;
NUMREDS  = 1;  LOOP     = 0;
SETTING  = UNFILLED;

A = zeros(SIZE,SIZE,SIZE);
A(X,Y,Z) = RED;
LEFT    = X - 1;  RIGHT   = X + 1;
TOP     = Y - 1;  BOTTOM  = Y + 1;
FRONT   = Z - 1;  BACK    = Z + 1;
ASSIGNLEFT;       ASSIGNRIGHT;
ASSIGNTOP;        ASSIGNBOTTOM;
ASSIGNFRONT;         ASSIGNBACK;
LEFT1   = LEFT;      RIGHT1  = RIGHT;
TOP1    = TOP;       BOTTOM1 = BOTTOM;
BACK1   = BACK;      FRONT1  = FRONT;

for p = 1:SIZE
   for q = 1:SIZE
      A(p,SIZE-6,q) = BLACK;
      A(p,SIZE-11,q) = BLACK;
   end
end

for LOOP = 1:CELLS;
   CHOOSE3d2;
   SETRED;
end
VIEWTUMOR;
TOTALREDS;
B=NUMREDS;
```

VIEWTUMOR.m See Tumor3d.

MATLAB Code for Tumor3d3

ASSIGNBACK.m See Tumor3d.

ASSIGNBACK1.m See Tumor3d.

ASSIGNBOTTOM.m See Tumor3d.

ASSIGNBOTTOM1.m See Tumor3d.

ASSIGNFRONT.m See Tumor3d.

ASSIGNFRONT1.m See Tumor3d.

ASSIGNLEFT.m See Tumor3d.

ASSIGNLEFT1.m See Tumor3d.

ASSIGNRIGHT.m See Tumor3d.

ASSIGNRIGHT1.m See Tumor3d.

ASSIGNTOP.m See Tumor3d.

ASSIGNTOP1.m See Tumor3d.

CHECKDIR.m

```
%When the tumor hits the edge of the growth space, the division is
stopped and a new
%direction must be chosen.  The "CUM", or cumulative, direction values
represent the
%collection of the "delta" direction values as a cell divides over a
direction.  When
%zero, the dividing cell is one cell space (delta distance) away from
the parent cell.
if i+ROWCUM+DELTAROW < 1 | j+COLCUM+DELTACOL < 1 | k+DEPCUM+DELTADEP <1
   DIRECTION;
   CHECKDIR;
%When the tumor hits the edge of the growth space
elseif i+ROWCUM+DELTAROW > SIZE | j+COLCUM+DELTACOL > SIZE |
k+DEPCUM+DELTADEP > SIZE
   DIRECTION;
   CHECKDIR;
%If the cell space is available (BLUE), designate it YELLOW, determine
its distance
%for inner matrix calculations, and change SETTING to FILLED.
elseif A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL,k+DEPCUM+DELTADEP) == BLUE
   A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL,k+DEPCUM+DELTADEP) = YELLOW;
   ABS1 = abs((i+ROWCUM+DELTAROW)-X);
   ABS2 = abs((j+COLCUM+DELTACOL)-Y);
   ABS3 = abs((k+DEPCUM+DELTADEP)-Z);
   VECTOR = [ABS1,ABS2,ABS3];
   ABSOLUTE = max(VECTOR);
   SETTING = FILLED;
%If the cell is a baundary cell (BLACK), then subract a hit point and
choose a new direction.
%Since the "CUM", or cumulative, direction values are not manipulated,
the cell choosees a new
%direction focused on the cell space before the boundary cell, not at
the boundary.  Otherwise, the new direction
%could pass the dividing cell right throught the boundary.
elseif A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL,k+DEPCUM+DELTADEP) <=
BLACK & A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL,k+DEPCUM+DELTADEP)> RED
   A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL,k+DEPCUM+DELTADEP) =
A(i+ROWCUM+DELTAROW,j+COLCUM+DELTACOL,k+DEPCUM+DELTADEP) - 3;
   DIRECTION;
   CHECKDIR;
else
%Otherwise the cell is unavailable (RED or YELLOW).  So, add to the
delta values
%in the current direction
   ROWCUM = ROWCUM + DELTAROW;
   COLCUM = COLCUM + DELTACOL;
```

```
   DEPCUM = DEPCUM + DELTADEP;
   CHECKDIR;
end
```

CHOOSE3d3.m See Tumor3d.

DIRECTION.m See Tumor3d2.

SETRED.m See Tumor3d.

TOTALREDS.m See Tumor3d.

Tumor3d3.m

```
function B = Tumor3d3(SIZE,CELLS,X,Y,Z)

%Initialize variables
DELTAROW = 0;   DELTACOL = 0;   DELTADEP = 0;
ROWCUM   = 0;   COLCUM   = 0;   DEPCUM   = 0;
LEFT     = 0;   RIGHT    = 0;
TOP      = 0;   BOTTOM   = 0;
FRONT    = 0;   BACK     = 0;
FILLED   = 1;   UNFILLED = 0;
BLACK    = 27;  RED      = 2;
YELLOW   = 1;   BLUE     = 0;
ABSOLUTE = 0;   ABSMAX   = 0;
NUMREDS  = 1;   LOOP     = 0;
SETTING  = UNFILLED;

%Set up cube of zeros
A = zeros(SIZE,SIZE,SIZE);
%Assign initial malignant cell to RED (User defined position)
A(X,Y,Z) = RED;
%Define "Inner Matrix": This cuts down on the computational
%time by allowing the algorithm to search for RED cells only within
%the Inner Matrix, which is one cell level away from all existing
%RED cells.
LEFT    = X - 1;  RIGHT   = X + 1;
TOP     = Y - 1;  BOTTOM  = Y + 1;
FRONT   = Z - 1;  BACK    = Z + 1;
ASSIGNLEFT;    ASSIGNRIGHT;
ASSIGNTOP;     ASSIGNBOTTOM;
ASSIGNFRONT;      ASSIGNBACK;
%The Left1, Right1, etc. define a YELLOW inner matrix that updates
%within the CHOOSE3d3 algorithm.  The YELLOW inner matrix keeps up with
the
%changing state of the dividing tumor by drawing a matrix around
%all newly created YELLOW cells.  Doing this keeps the algorithm from
%checking for RED cells in an area only occupied by new YELLOW cells.
%Thus when the CHOOSE3d3 algorithm is finished, the RED Inner Matrix
%is assigned to the value of the YELLOW inner matrix.
LEFT1   = LEFT;      RIGHT1  = RIGHT;
TOP1    = TOP;       BOTTOM1 = BOTTOM;
BACK1   = BACK;      FRONT1  = FRONT;
```

```
%Creates a duct-like Boundary
J = 5;
for I=1:SIZE
   A(J,J+5:J+11,I)     = BLACK;
   A(J+1,J+4:J+5,I)    = BLACK;  A(J+1,J+11:J+12,I)  = BLACK;
   A(J+2,J+3:J+4,I)    = BLACK;  A(J+2,J+12:J+13,I)  = BLACK;
   A(J+3,J+2:J+3,I)    = BLACK;  A(J+3,J+13:J+14,I)  = BLACK;
   A(J+4,J+1:J+2,I)    = BLACK;  A(J+4,J+14:J+15,I)  = BLACK;
   A(J+5,J+1,I)        = BLACK;  A(J+5,J+15,I)       = BLACK;
   A(J+6,J+1,I)        = BLACK;  A(J+6,J+15,I)       = BLACK;
   A(J+7,J:J+1,I)      = BLACK;  A(J+7,J+15:J+16,I)  = BLACK;
   A(J+8,J,I)          = BLACK;  A(J+8,J+16,I)       = BLACK;
   A(J+9,J:J+1,I)      = BLACK;  A(J+9,J+15:J+16,I)  = BLACK;
   A(J+10,J+1,I)       = BLACK;  A(J+10,J+15,I)      = BLACK;
   A(J+11,J+1,I)       = BLACK;  A(J+11,J+15,I)      = BLACK;
   A(J+12,J+1:J+2,I)   = BLACK;  A(J+12,J+14:J+15,I) = BLACK;
   A(J+13,J+2:J+3,I)   = BLACK;  A(J+13,J+13:J+14,I) = BLACK;
   A(J+14,J+3:J+4,I)   = BLACK;  A(J+14,J+12:J+13,I) = BLACK;
   A(J+15,J+4:J+5,I)   = BLACK;  A(J+15,J+11:J+12,I) = BLACK;
   A(J+16,J+5:J+11,I)  = BLACK;
end

%Algorithm for dividing
for LOOP = 1:CELLS;
   CHOOSE3d3;
   SETRED;
end
%Change the value of all BLACK cells so that they can be seen through
CHANGEBLACK;
%Creates images of face densities
VIEWTUMOR;
%Count total number of malignant cells
TOTALREDS;
%Return this number
B=NUMREDS;
```

VIEWTUMOR.m

```
%Clear the existing image in figure(x) storage
clear figure(1)
clear figure(2)
clear figure(3)
%Sum the values of each cell across the respective cube face
C=sum(A,1);              %Side View
F=sum(A,2);              %Overhead View
H=sum(A,3);              %Front View

%Change the Sums of the first two faces to matrices that
%can be used by the image() command
l = 0;
m = 0;
D=zeros(SIZE);
for l = 1:SIZE
   D = cat(1,D,C(:,:,l));
```

```
end
E = D(SIZE+1:2*SIZE,1:SIZE);

G=F(:,:,1);
for m = 2:SIZE
   G = cat(2,G,F(:,:,m));
end

%Display the images for each viewpoint
figure(1)
hold on;
image(uint8(E));
title('Side View')
hold off;

figure(2)
hold on;
image(uint8(G));
title('Overhead View')
hold off;

figure(3)
hold on;
image(uint8(H));
title('Front View')
hold off;
```

Appendix C – Contacts and Internet Search Information.

Much of the work completed on this thesis required a firm understanding of how breast cancer behaves, how mammography works, and how the problem may be solved. Listed below are people or organizations who added incredible insight to this thesis research effort and how to contact them.

Table C-1. Contacts.

Name	Organization	Phone	E-mail
Dr. Kenneth Bauer	Dept of Operations Research	(937) 255-6565 x4328	Kenneth.Bauer@afit.af.mil
Lt Col J.O. Miller	Dept of Operations Research	(937) 255-6565 x4326	John.Miller@afit.af.mil
Major Del Wilson	Air Force Research Laboratory	(937) 255-3122	DEL.WILSON@FALCON.WPAFB.AF.MIL
Dr. Steve Rogers	Qualia Computing	(937) 431-1464	capt_amerika@qualia-computing.com
Dr. Jeff Hoffmeister	Qualia Computing	(937) 431-1464	Jeff_Hoffmeister@qualia-computing.com
Dr. Matthew Kabrisky	Dept of Electrical Engineering	(937) 255-3636 x4541	mkabrisk@afit.af.mil
Phil Amburn	Qualia Computing	(937) 431-1464	N/A
Dr. Rebecca Glaser	Regional Hospital Physician	N/A	N/A
Dr. Andy Chunn	Miami Valley Hospital	N/A	N/A

In preparation for the literature review and medical knowledge needed for this research effort, I conducted several Internet subject searches. The first is a series of searches on breast cancer basics. The second focuses on the keyword "microcalcifications". Finally, the last search focused on the visualizations aspect of breast cancer, such as slides of tumor cross-sections, mammograms, and information on a JAVA applet that allows 3-dimensional structures to be rotated. The following is the list of Internet web sites that I researched:

Internet Search on General Breast Cancer Topics

1. **American Institute for Cancer Research.** (www.aicr.org) – Lists a number of grants associated with various cancers. These research grants deal mainly with the effect of various chemicals, diets, and elements (selenium, vitamins, etc.) upon cancer. Focuses on information concerning the "two major influencers of cancer risk", diet and smoking.
2. **Biology of the Mammary Gland Website.** (mammary.nih.gov) – Sponsored by the National Institutes of Health, Bethesda, MD. Possibly the only website that touches, in detail, the structure of the breast and the associated tumors. The language, however, is highly clinical. Most of the research seems to be based on using mice as models. Pictures of breast cells and a comparative pathology mouse and human mammary glands are included. A self-guided tutorial of Powerpoint slides is included.
3. **BreastCancer.Net.** (www.breastcancer.net) – An information site with various article listings, some on screening and mammography imaging and the problems associated with it. Links to Research organizations and Ongoing Studies.
4. **Center for Advancement in Cancer Research.** (www.lifeenrichment.com) - Their project is aimed at uncovering possible new approaches to preventing breast cancer and other cancers in women. Possible site for further documentation that must be ordered.
5. **University of Pennsylvania Cancer Center.** (oncolink.upenn.edu) – An excellent site that covers screening, tumor classification, and general information about breast cancer. This site has a "virtual classroom" that tours through the site and various lectures.
6. **Lawrence Berkeley National Laboratory ELSI Project.** (www.lbl.gov/Education /ELSI/screening-main.html) – Very easy to follow questions. Offers easy to understand reports of the latest reports on breast cancer screening advancements and news.
7. **National Cancer Institute – FCRDC.** (www.ncifcrf.gov/fcrdc/resources/index.html) – Links to technical resources.
8. **National Cancer Institute (NCI).** (www-dceg.ims.nci.nih.gov/conference/ agenda.html) – A recent conference yielded some interesting outcomes. Abstracts on studies can be read. One Dr. Gail Geller (John Hopkins Univ.) states that "at the current time, genetic testing for breast cancer susceptibility is not likely to improve either primary or secondary efforts to prevent breast cancer because of limitations in the efficacy (benefit) of preventive interventions".

9. **Advanced Research Center (ARC).** (www.arc.com/cancer/cancernet.html), (198.17.244.10/database/Cancernet/english/hpsupport/index/html) – Enables easy access to NCI's database. Breast cancer screening and prevention information for physicians and patients. Contains randomized mammography results from several countries.
10. **Medicine Online.** (www.meds.com) – Articles concerning breast cancer tumor modeling. A query can be done to find applicable articles. Some articles mention size, shapes, and various tumor characteristics. Could be very useful, despite the clinical language.
11. **Beth Israel Health Care System.** (www.wp.com/bicbs/gtoc.html) – Patient's guide to breast cancer. Possibly the most informative introduction to breast cancer. Well laid-out.

Various Sites Checked:

12. Breast Cancer Resource Center of Austin. (www.bcrc.org) – Information center with links.
13. Susan G. Komen Breast Cancer Foundation. (www.breastcancerinfo.com) – Raising money for breast cancer.
14. American College of Surgeons Commission on Cancer. (www.facs.org) – Information on the Commission on Cancer.
15. Scientific American. (www.sciam.com/0996issue/0996breast.html) – Good introduction to breast cancer in general with figures.
16. Community BreastHealth Project. (www-med.stanford.edu/CBHP) – Links.
17. Cancer Research Foundation of America. (www.crfa.org) – Grants and community news.
18. Fred Hutchinson Cancer Research Center. (www.fhcrc.org) – Technical shared resources.
19. Dr. Jankharia's Imaging Centre. (www.jankharia.com) – Small excerpt on X-ray history and mammography general info.
20. Jim Kenzig's General Cancer Links. (www.virtualtrials.com/btlinks/cancer.html) – About 35 links on different cancers, support groups, information centers.
21. Sloan-Kettering Cancer Center. (www.mskcc.org/lr.htm) – Research and education programs run through the center. Includes reports on prevention and control of various cancers.
22. National Alliance of Breast Cancer Organizations. (NABCO). (www.nabco.org) – Trial and resource information. Allows browser to enter into clinical trials.
23. Harvard Medical School. (www.hms.harvard.edu/research.html) – Division of school.

Internet Search Results on "Microcalcifications"

1. http://www-dsed.llnl.gov/documents/imaging/jmhspie93.html
2. http://www.seas.gwu.edu/student/tlooms/MGT243/prof_refs2.html
3. http://www.rose.brandeis.edu/users/mammo/digital.html
4. http://www.bme.unc.edu/~mood/mammo.html
5. http://www.pslgroup.com/dg/2c3fe.htm

Internet Search Results for Visualization Aspects

1. http://www-medlib.med.utah.edu/WebPath/BRESHTML/BREST049.html

2. http://www.tumorboard.com/cgi-bin/dsearch.cgi
3. http://marathon.csee.usf.edu/Mammography/DDSM/thumbnails/normal.html
4. http://www.medscape.com/Medscape/features/ImageofWeek/public.html
5. http://www.uhrad.com/Default.html
6. http://dms2.nlm.nih.gov/radocs/neoii/neo2naud.htm
7. http://er4www.eng.ohio-state.edu/~siegr/zincblen.htm

Bibliography

Adam, John Anthony and Nicola Bellomo, Eds. A Survey of Models for Tumor Immune Systems Dynamics. Boston: Birkhauser, 1997.

Aldaz, C. Marcelo, et al, Ed. "Etiology of breast cancer and gynecological cancers". Proceedings of the Ninth Annual International Conference on Carcinogenesis and Risk Assessment held at Austin, Texas. New York: Wiley-Liss, 1997.

American Medical Women's Association. Breast Cancer Education for DoD Primary Care Managers: Participant Manual. Alexandria: 1996.

Beth Israel Health Care System. "Patients Guide to Breast Cancer." Excerpt from unpublished article, n. pag. http://www.wehealny.org/healthinfo/breastcancer/ubc.html. 16 February 1999.

Brunner, S., B. Langfeldt, P.E. Andersen, editors. Early Detection of Breast Cancer. Berlin: Springer-Verlag, 1984.

Ceriani, Roberto L., Ed. "Antigen and antibody molecular engineering in breast cancer diagnosis and treatment". Proceedings of the International Workshop on Breast Cancer Research Conference. New York: Plenum Press, 1994.

Connelly, Pat. "Breast Cancer." Slide form unpublished article, n. pag. http://www.erinet.com/fnadoc/brest.htm. 17 February 1999.

Cotran, Ramzi S., Vinay Kumar, and Stanley L. Robbins. Robbins Pathologic Basis of Disease (Fourth Edition). Philadelphia: W.B. Saunders Company, 1989: 1192-1201.

Craig, John A. Breast Lumps. Summit, NJ: The Division, 1980 [slides].

Harris, Jay R., Samuel Hellman, I. Craig Henderson, and David W. Kinne. Breast Diseases (Second Edition). Philadelphia: J.B. Lippincott Company, 1991.

Klatt, Edward C. "Breast Pathology Index.", Index of slides, n. pag. http://www-medlib.med.utah.edu/WebPath/BRESHTML/BRESTIDX.html#2. 16 February 1999.

Kopans, Daniel B. Breast Imaging. Philadelphia: Lippincott-Raven, 1998: 29-37.

Olivotto, Ivo, Karen Gelmon, and Urve Kuusk. Intelligent Patient Guide to Breast Cancer. Vancouver: Intelligent Patient Guide Ltd., 1995.

Parker, Steve H. and William E. Jobe. Percutaneous Breast Biopsy. New York, NY: Raven Press, 1993.

Porrath, Saar. A Multimodality Approach to Breast Imaging. Rockville, MD: Aspen Publishers, Inc., 1986.

Schroeder, Will, Ken Martin and Bill Lorensen. The Visualization Toolkit. Upper Saddle River, New Jersey: Prentice Hall PTR, 1996: 146-154, 331-429.

Smathers, Ralph L. "TF Case: Craniocaudal View Positioning Failure." Slide from article, n. pag. http://mammo.net/m1008e.htm. 17 February 1999.

University Hospitals of Cleveland. "Women's Diagnostic Imaging Teaching Files". Mammogram images and reports. http://www.uhrad.com/mamarc.htm. 23 February 1999.

University of Pennsylvania. "What is Breast Cancer?" Excerpt from unpublished article, n. pag. http://oncolink.upenn.edu/pdq_html/2/engl/200013.html. 16 February 1999.

University of South Florida. "Digital Database for Screening Mammography." Slides from database, n. pag., http://marathon.csee.usf.edu/Mammography/DDSM/thumbnails/normals/normal_01/case0069/A-0069-1.html. 17 February 1999.

Zander, J. and J. Baltzer, editors. Early Breast Cancer: Histopathology, Diagnosis, and Treatment. Berlin: Springer-Verlag, 1985.

Vita

First Lieutenant Christopher Brian Bassham was born on 28 December 1972 in Columbia, Tennessee. He graduated from Richland High School in Lynnville, Tennessee in May 1991. He entered undergraduate studies at the United States Air Force Academy in Colorado Springs, Colorado where he graduated with a Bachelor of Science degree in Operations Research in May 1995 and was commissioned as an Air Force officer.

His first assignment was with the 49th Test Squadron at Barksdale Air Force Base as a flight test engineer/systems analyst in August 1995. In August 1997, he entered the Operational Analysis program, School of Engineering, Air Force Institute of Technology. Upon graduation, he will remain at the Air Force Institute of Technology where he will extend into the Operational Sciences doctorate program.

Permanent Address: 1910 Big Dry Creek Road
Pulaski, TN 38478

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March, 1999	Master's Thesis

4. TITLE AND SUBTITLE
Visualizing Early-Stage Breast Cancer Tumors in a Mammographic Environment Through a 3-Dimensional Mathematical Model

5. FUNDING NUMBERS

6. AUTHOR(S)
Christopher Brian Bassham, 1st Lieutenant, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
2950 P Street
WPAFB, OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOA/ENS/99M-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Major Delano Wilson
Air Force Research Laboratory
1864 4th Street
WPAFB, OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
In response to the insidious and deadly nature of breast cancer and the less-than-perfect detection ability of mammography, we develop a mathematical model as a foundation to the long-term goal of improving early breast cancer detection. By using modeling and simulation to construct an accurate breast cancer tumor model, we hope to solve the problems associated with mammogram misdiagnosis and, perhaps as a by-product, lend insight to tumor development dynamics. The final tumor model, written in MATLAB, provides realistic tumor growth and 2-dimensional visualization of 3-dimensional structures. Earlier modeling attempts capture slices of the tumor in the 2-dimensional growth spaces. The final 3-dimensional model closely mimics the characteristics of theoretical breast cancer development within the female breast by establishing an algorithm that reliably represents the ideal tumor model. The possible impact of this model and its progeny is earlier detection of breast cancer, which leads to an increased chance of survival for those afflicted with the disease.

14. SUBJECT TERMS
Breast Cancer; tumor; modeling; MATLAB; visualization; simulation; growth sequence; mammography; x-rays; 3-dimensional; algorithm

15. NUMBER OF PAGES
111

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSSIFIED	UNCLASSSIFIED	UNCLASSSIFIED	UL